

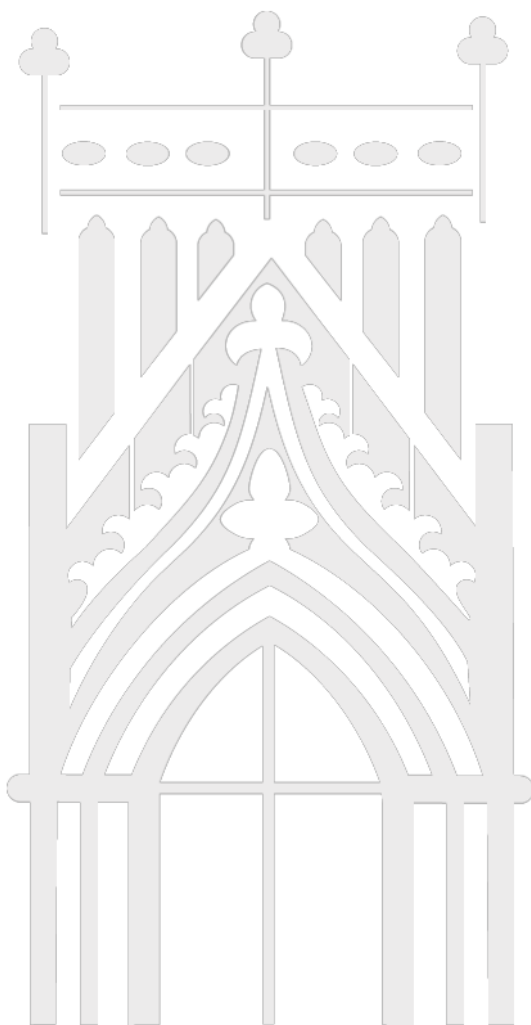
Mestrado em Computação Móvel

Help Medicine

Sistema de alerta de toma de medicação

Anabela Pais Tavares

julho | 2019



Escola Superior
de Tecnologia e Gestão



Help Medicine

Sistema de alerta de toma de medicação

Relatório de Projeto Aplicado submetido como requisito parcial para a
obtenção do grau de
Mestre em Computação Móvel

Orientador: Professora Doutora Natália Fernandes Gomes
Coorientador: Professor Doutor Noel de Jesus Mendonça Lopes

Anabela Pais Tavares

março | 2019

*“A maior recompensa para o trabalho do homem não é o que ele ganha
com isso, mas o que ele se torna com isso.”*

Jonh Ruskin

Agradecimentos

Com a entrega desta dissertação vejo chegar ao fim dois anos de um percurso acadêmico que sempre desejei e que realizam a minha vida tanto a nível pessoal como profissional.

A concretização desta etapa não seria possível sem o esforço notório de uma vida de duro trabalho por parte dos meus pais. Aos meus pais o meu eterno e mais sincero agradecimento.

Ao meu irmão, cunhada e tia, pela motivação de todo este projeto, obrigada pelas palavras sábias, pela experiência de vida partilhada e por todo o amor que vejo nos vossos sorrisos.

Aos meus orientadores, Professora Natália Gomes e Professor Noel Lopes, por todo o apoio e por me ofertarem com grandes ideias ao longo de todo o desenvolvimento do projeto. O meu muito obrigado pela atenção e disponibilidade que sempre mostraram ter.

A todos os meus amigos, por estarem sempre presentes, em especial aqueles que me acompanharam durante estes dois anos de formação académica, pelas noites de trabalho e de estudo e por partilharem comigo as melhores experiências de uma vida académica.

E por último, mas não menos importante, ao meu namorado, por me acompanhar ao longo destes anos, por me dirigir sempre as melhores e mais reconfortantes palavras e por toda a paciência, compreensão e confiança que sempre demonstrou nas minhas capacidades.

Resumo

O número de pessoas a tomar medicação diária aumentou consideravelmente nos últimos 40 anos. A grande maioria da população é polimedicada e muitas vezes esquece-se da toma dos medicamentos, por isso, a gestão de medicamentos torna-se fulcral para proporcionar uma melhoria da qualidade de vida destes [1]. A sobrecarga das dosagens e os diferentes horários de toma fazem com que os pacientes se confundam no cumprimento do esquema aconselhado pelo médico. Torna-se, portanto, fundamental o desenvolvimento de sistemas inteligentes que auxiliem o paciente na gestão da sua medicação.

A presente dissertação de mestrado foi materializada numa aplicação móvel e num sistema embebido, designado *Help Medicine*, que visa responder aos problemas da polimedicação ou simplesmente o esquecimento de tomar a medicação, através de uma solução tecnológica que incorpora as vertentes de controlo e comunicação. O objetivo do *Help Medicine* é relembrar, de forma automática, o paciente da toma atempada dos seus medicamentos e consiste numa aplicação móvel híbrida e num protótipo de um dispositivo com várias partições para organização dos vários medicamentos. Este foi pensado para utilizadores que não tinham acesso a um *smartphone* e, por isso, os sistemas não comunicam entre si.

A aplicação móvel regista dados do paciente, da toma de medicação (data, hora e nome do medicamento) e pedido de encomenda de medicamentos, através de email, à farmácia que o paciente pretenda. Também, é possível a verificação de tomas anteriores e uma visão geral de tomas por mês. Aquando a chegada da data e hora da toma de cada medicação, a aplicação alerta o paciente através de uma notificação sonora e/ou vibratória no *smartphone*.

O sistema embebido apresenta 7 compartimentos que dizem respeito a pelo menos uma toma de medicação por cada dia da semana. Os compartimentos em questão devem ser preenchidos com a devida medicação, pelo paciente ou pelo seu cuidador, no início de cada semana e definidos os alarmes necessários para cada toma. Na data e hora

da toma de cada medicação, o sistema desenvolvido emite um lembrete ao paciente através de um alerta luminoso e sonoro. O sistema só desliga os alertas quando o paciente se aproximar do dispositivo. Os testes de validação realizados ao *Help Medicine* revelaram que o protótipo se mostra funcional e apto para integrar o quotidiano de qualquer paciente.

Palavras-chave: Dispensador de Medicamentos; Sistema embebido; Aplicações móveis; Controlo de toma de medicamentos;

Abstract

In the last 40 years the number of people taking daily medication has increased considerably. The vast majority of the population is polymedicated and often forgotten, and therefore drug management becomes central to provide an improvement in their quality of life [1]. The overload of the dosages and the different times of taking medication cause the patients to be confused regarding the fulfilment of the scheme advised by the doctor. It is therefore fundamental to develop intelligent systems that assist the patient managing his medication.

This dissertation was materialized in a mobile application and in an embedded system called *Help Medicine*, which aims to respond to the problems of polymedication or simply on forgetting to take the medication through a technological solution that incorporates the control and communication aspects. The purpose of *Help Medicine* is to automatically remind the patient of the timely intake of their medications and consists of a hybrid mobile application and a prototype of a multi-compartment device for organizing diverse medications. The later is intended for users that do not have access to a smartphone, and therefore the systems don't communicate with each other.

The mobile application registers the patient's data, medication intake (date, time and name of the medication) and it orders the medication by email to the pharmacy chosen the patient. Also, it is possible to check previous shots and to get a monthly overview. Upon arrival of the date and time of each medication, the application alerts the patient through a sound and with or without vibration notification on the smartphone.

The embedded system presents 7 compartments that correspond to a take of medication per each day of the week. The compartments in question must be fulfilled with medication, either by the patient or by its caregiver. At the beginning of each week they define the alarms required for each dose. At the time and date of each medication, the developed system sends a reminder to the patient using a luminous and audible alert. The system will only turn off alerts when the patient approaches to the device. The

validation tests performed at *Help Medicine* revealed that the prototype is functional and able to integrate the daily life of any patient.

Keywords: Medication Dispenser; Embedded system; Mobile applications; Assisted Living Environment; Medication in the present day;

Índice

Agradecimentos	i
Resumo	iii
Abstract	v
Índice	vii
Índice de figuras	ix
Índice de tabelas	xiii
Lista de abreviaturas e acrónimos	xv
1. Introdução	17
1.1. Contextualização e motivação	17
1.2. Objetivos	19
1.3. Estrutura da dissertação	20
2. As tecnologias e a saúde	23
2.1. A evolução das tecnologias	23
2.1.1. O impacto da tecnologia na sociedade	24
2.1.2. O impacto da tecnologia na saúde	26
2.2. Soluções tecnológicas no âmbito do projeto	27
2.2.1. Soluções sobre saúde informativa	28
2.2.2. Soluções sobre medicação	30
3. Metodologia, planeamento e análise de requisitos	35
3.1. Metodologia	35
3.2. Planeamento	36
3.3. Análise de requisitos da aplicação móvel	39
3.3.1. Técnicas de recolha e levantamento de requisitos	39
3.3.2. Requisitos funcionais e não funcionais	41

3.3.3.	Desenho e modelação	44
3.4.	Análise de requisitos do sistema embebido.....	64
3.4.1.	Protótipo do sistema de alerta de medicamentos.....	65
3.4.2.	Fluxograma do sistema de alerta de medicamentos	68
4.	Desenvolvimento e implementação <i>Help Medicine</i>	71
4.1.	Desenvolvimento da aplicação <i>Help Medicine</i>	71
4.1.1.	Desenvolvimento do <i>webservice</i>	73
4.1.2.	Desenvolvimento do <i>frontend</i>	79
4.2.	Desenvolvimento do sistema embebido de alerta de medicamentos.....	83
4.2.1.	Arquitetura do sistema embebido <i>Help Medicine</i>	84
4.2.2.	Módulos do dispositivo <i>Help Medicine</i>	85
5.	Testes e resultados.....	91
5.1.	Validação do sistema <i>Help Medicine</i>	91
5.1.1.	Características da amostra.....	91
5.1.2.	Resultados e Discussão	93
6.	Conclusões.....	95
6.1.	Trabalhos Futuros.....	96
	Bibliografia.....	99
	Anexo A – Análise de requisitos	103
	Descrição casos de uso e diagramas de sequência.....	103
	Anexo B – <i>Wireframes</i>	123
	Anexo C – Formulário de validação do <i>Help Medicine</i>	125

Índice de figuras

Figura 1 – Ecrã principal da <i>WebMD</i>	28
Figura 2 – Ecrã identificar doenças através dos sintomas.....	28
Figura 3 – Ecrã listagem de medicamentos.....	28
Figura 4 - Sistema <i>MySignals</i> [19]	29
Figura 5 – <i>Cloud</i> do <i>MySignals</i> [19].....	30
Figura 6 – Visualização do alerta do medicamento da aplicação <i>MediSafe</i>	31
Figura 7 – Ecrã principal da aplicação <i>MediSafe</i>	31
Figura 8 – <i>Widget</i> da aplicação <i>Medisafe</i>	32
Figura 9 - <i>Feed de notícias</i> da aplicação <i>MediSafe</i>	32
Figura 10 – Dispensador <i>MedMinder</i> – Maya [23]	33
Figura 11 - Dispensador <i>MedMinder</i> – Jon [23]	33
Figura 12 - Percentagem de tempo gasto por atividade	37
Figura 13 – Etapas do planeamento apresentadas através do diagrama de <i>Gantt</i>	38
Figura 14 - Diagrama de contexto da aplicação <i>Help Medicine</i>	46
Figura 15 – Diagrama de casos de uso da aplicação <i>Help Medicine</i>	47
Figura 16 – Sequência de operações – “Criar Utilizador”	52
Figura 17 - Sequência de operações – “Adicionar alerta de medicamentos”.....	55
Figura 18 - Sequência de operações – “Inserir encomenda”.....	58
Figura 19 - Sequência de operações – “Notificação de alerta medicamento”.....	60
Figura 20 – Diagrama de classes da aplicação <i>Help Medicine</i>	62
Figura 21 – <i>Wireframe</i> da página inicial do sistema <i>Help Medicine</i>	63
Figura 22 – <i>Wireframe</i> da página alerta de medicamentos do sistema <i>Help Medicine</i> ..	64
Figura 23 - Desenho do protótipo do sistema de alerta de medicamentos <i>Help Medicine</i> em <i>CorelDraw</i>	66
Figura 24 – Protótipo final do sistema de alerta de medicamentos <i>Help Medicine</i>	67
Figura 25 – Fluxograma do sistema de alerta de medicamento <i>Help Medicine</i>	69
Figura 26 – Arquitetura do sistema da aplicação móvel <i>Help Medicine</i>	73
Figura 27 – Pedido para ver todos os alertas de medicamentos existentes.	74
Figura 28 – Inserção de um novo alerta de medicamento.....	76

Figura 29 – Exemplo de um pedido de alteração de dados	76
Figura 30 – Exemplo de um pedido de apagar dados	77
Figura 31 – Exemplo de um <i>email</i> enviado pelo <i>webservice</i> à farmácia	77
Figura 32 – Configuração do <i>EmailAlert</i> no <i>Salesforce</i>	78
Figura 33 – Desenvolvimento do algoritmo para envio de <i>emails</i>	79
Figura 34 – Ecrã inicial da aplicação <i>Help Medicine</i>	80
Figura 35 – Ecrã dos alertas criados para a toma de medicação	81
Figura 36 – Ecrã de registo do alerta de toma de medicação	81
Figura 37 – <i>Auto-complete</i> do ecrã de registo do alerta.....	82
Figura 38 – Notificação de alerta ao paciente para tomar a medicação.....	82
Figura 39 – Ecrã detalhes das encomendas do paciente	83
Figura 40 – Ecrã encomenda de medicamentos.	83
Figura 41 – Arquitetura do sistema embebido	84
Figura 42 – Diagrama de blocos do sistema embebido	85
Figura 43 – Representação <i>push button</i> utilizado no projeto [38].....	86
Figura 44 – Representação esquemática de um LED [40].....	86
Figura 45 - Representação de um LCD adaptável para utilização no Arduíno [42].....	87
Figura 46 – Representação do <i>Buzzer</i> utilizado no projeto [43].....	88
Figura 47 – Representação de um Sensor infravermelho utilizado no projeto	88
Figura 48 – Caixa de Medicamentos semanal, utilizada no projeto	89
Figura 49 - Sequência de operações – “Editar Utilizador”.	105
Figura 50 - Sequência de operações – “Editar alerta medicamentos”.	107
Figura 51 - Sequência de operações – “Eliminar alerta medicamentos”	109
Figura 52 - Sequência de operações – “Consultar alerta medicamento”	111
Figura 53 - Sequência de operações – “Inserir medicamento”.....	113
Figura 54 - Sequência de operações – “Consultar medicamentos”	115
Figura 55 - Sequência de operações – “inserir Farmácia”	117
Figura 56 - Sequência de operações – “Consultar farmácias”	119
Figura 57 - Sequência de operações – “Consultar encomendas”	121
Figura 58 – <i>Wireframe</i> utilizadores da aplicação <i>Help Medicine</i>	123
Figura 59 - <i>Wireframe</i> inserir utilizadores da aplicação <i>Help Medicine</i>	123
Figura 60 - <i>Wireframe</i> menu dos utilizadores da aplicação <i>Help Medicine</i>	123

Figura 61 - <i>Wireframe</i> perfil do utilizador da aplicação <i>Help Medicine</i>	123
Figura 62 - <i>Wireframe</i> inserir alerta da aplicação <i>Help Medicine</i>	124
Figura 63 - <i>Wireframe</i> inserir medicamento da aplicação <i>Help Medicine</i>	124

Índice de tabelas

Tabela 1 – Requisitos Funcionais da aplicação <i>Help Medicine</i>	42
Tabela 2 – Requisitos não funcionais da aplicação <i>Help Medicine</i>	43
Tabela 3 – Requisitos funcionais do sistema embebido	43
Tabela 4 – Requisitos não funcionais do sistema embebido.	43
Tabela 5 – Descrição do caso de uso “Criar utilizador”	50
Tabela 6 - Descrição do caso de uso “Adicionar alerta medicamentos”	53
Tabela 7 - Descrição do caso de uso “Inserir encomenda”	56
Tabela 8 - Descrição do caso de uso “Notificação alerta medicamento a utilizar ao utilizador”	59
Tabela 9 - Detalhes da amostra da população	92
Tabela 10 – Descrição do caso de uso “Editar utilizador”	103
Tabela 11 - Descrição do caso de uso “Editar alerta medicamentos”	106
Tabela 12 - Descrição do caso de uso “Eliminar alerta medicamento”	108
Tabela 13 - Descrição do caso de uso “Consultar alerta medicamento”	110
Tabela 14 - Descrição do caso de uso “Inserir medicamento”	112
Tabela 15 - Descrição do caso de uso “Consultar medicamentos”	114
Tabela 16 - Descrição do caso de uso “Inserir Farmácias”	116
Tabela 17 – Descrição do caso de uso “Consultar farmácias”	118

Lista de abreviaturas e acrónimos

XP	<i>eXtreme Programming</i>
TIC	Tecnologias da Informação e Comunicação
IFTF	<i>Institute For The Future</i>
LED	<i>Light Emitting Diode</i>
LCD	<i>Liquid crystal display</i>
GSM	<i>Global System for Mobile Communications</i>
VORD	<i>View point Oriented Requirements Definition</i>
UML	<i>Unified Modeling Language</i>
DFD	Diagrama de fluxo de dados
UUID	<i>Unique Identifier</i>
PNG	<i>Portable Network Graphics</i>
JSON	<i>JavaScript Object Notation</i>
HTTP	<i>Hypertext Transfer Protocol</i>
POO	Programação orientada a objetos
DML	<i>Data Manipulation Language</i>
URL	<i>Uniform Resource Locator</i>

1. Introdução

O presente relatório de projeto apresenta a análise e modelação de dois sistemas (hardware e Software) no âmbito das tecnologias e a saúde. Este projeto aplicado surge no âmbito do Mestrado em Computação Móvel, do Instituto Politécnico da Guarda. O principal objetivo é o de desenvolver sistemas que permitam alertar pacientes da toma dos seus medicamentos.

1.1.Contextualização e motivação

Graças à evolução da ciência e em particular da área da medicina, a expectativa de vida aumentou drasticamente [2]. Hoje em dia a população, nos países desenvolvidos tem ao seu alcance um conjunto de informação que lhe permite saber quais os melhores métodos para manter um estilo de vida mais saudável. O aparecimento dos antibióticos, a título de exemplo, tem provado ser uma ferramenta indispensável na luta contra as doenças, assim como o aparecimento de novos medicamentos cada vez mais eficazes [3]. Neste sentido, a indústria farmacêutica pode hoje ser considerada como uma das principais áreas de crescimento no que respeita à investigação, ao desenvolvimento e à inovação.

A par da ciência médica, podemos afirmar que a evolução das tecnologias cresceu de forma exponencial, sendo um exemplo a popularidade dos dispositivos móveis. Estes dispositivos, são cada vez mais utilizados tanto a nível pessoal como profissional. O uso de dispositivos móveis cresceu de tal forma, que as ferramentas associadas aos mesmos permitem aos seus utilizadores, entre inúmeras outras tarefas, fazer uma melhor gestão do seu dia-a-dia. Num mesmo modo, podemos afirmar que as tecnologias da informação são cada vez mais um aliado valioso para o bem-estar do ser humano [4].

A toma diária de medicação, pode tornar-se complicada uma vez que esta pode exigir a gestão de diferentes dosagens e em diferentes horários de toma, levando a que muitos pacientes necessitem de uma dependência de terceiros ou de assistência

permanecente para que deste modo os medicamentos sejam tomados de acordo com a prescrição médica [5]. Este problema pode colocar dificuldades na vida diária das pessoas, na gestão do seu dia-a-dia e afetar a sua vida pessoal e profissional.

Tendo em consideração alguns destes fatores, e percebendo as potencialidades que a tecnologia pode aportar à área da saúde muitas empresas têm tentado, de algum modo, que a tecnologia seja integrada cada vez mais nos cuidados de saúde [6]. A integração da tecnologia nesta área e o desenvolvimento de novas aplicações pode fornecer, a título de exemplo, uma solução que permita melhorar a qualidade de vida dos doentes através de sistemas que possibilitem de algum modo a alerta da toma de medicação.

Neste sentido, a presente dissertação tem como objetivo geral o desenvolvimento de uma solução tecnológica, denominado *Help Medicine*, que alerte os doentes que chegou a hora de tomar a medicação. O desenvolvimento deste projeto pretende contribuir para o desenvolvimento de novas soluções na área da tecnologia que permitam assim melhorar e facilitar o problema da gestão da medicação e consequentemente melhorar a qualidade de vida dos doentes.

O projeto *Help Medicine* é composto por duas soluções, uma aplicação móvel e um dispositivo eletrónico, que têm como principal objetivo, alertar o doente para a toma dos seus medicamentos.

A aplicação móvel é redirecionada para um publico mais jovem que tem acesso a smartphones mais facilmente. O funcionamento baseia-se numa arquitetura utilizador/servidor onde o utilizador através da aplicação móvel faz requisições ao servidor. O servidor processa o pedido e envia uma resposta com os dados para a aplicação móvel. Já o dispositivo eletrónico, foi desenvolvido para um público alvo mais idoso que não têm acesso a smartphones ou então não o sabe utilizar. O funcionamento deste é através da plataforma *open-source* Arduíno com os respetivos sensores que permitem que o utilizador seja alertado quando chegar a hora de tomar a medicação.

1.2. Objetivos

O principal objetivo deste projeto é o desenvolvimento de um sistema informático composto por duas soluções: uma aplicação móvel e um dispositivo eletrónico, tal como anteriormente referido, que permita alertar doentes da toma da sua medicação. As soluções, desenvolvidas no âmbito desta dissertação, vão permitir auxiliar os seus utilizadores na toma da medicação, contínua ou temporária, fornecendo deste modo uma solução para os doentes que não conseguem ou têm dificuldades de forma autónoma de realizar uma boa gestão da toma da sua medicação.

Os sistemas devem assim permitir, de um modo fácil, alertar o doente que este deve tomar a medicação no horário por ele estabelecido. De acordo com as duas soluções, apresenta-se os principais requisitos:

A aplicação móvel *Help Medicine* deve permitir:

- Registrar informações sobre o perfil do doente (nome, data de aniversário e *email*);
- Registrar a medicação do doente;
- Personalizar alerta para a toma de medicação;
- Efetuar a consulta dos alarmes que foram registados;
- Registrar encomendas de medicamentos;
- Efetuar consulta das encomendas.

No que respeita ao sistema embebido, este tem como objetivos permitir:

- Definir vários alarmes de acordo com o dia da semana;
- Alertar o doente aquando a hora da medicação;
- Desligar o alerta caso o doente já tenha tomado a medicação.

Para a concretização dos objetivos estabelecidos utilizou-se a metodologia *eXtreme Programming (XP)* que será descrita no Capítulo 3.

1.3.Estrutura da dissertação

A presente dissertação é composta por seis capítulos, sendo que o primeiro é o presente capítulo, onde para além de contextualizar o tema da dissertação, se apresentam as motivações que estiveram subjacentes à concretização dos objetivos, expondo-se, de forma breve, a metodologia utilizada no desenvolvimento do projeto.

O capítulo 2, sob o título *As tecnologias e a saúde*, apresenta uma breve exposição sobre a evolução da tecnologia, em particular na área da saúde evidenciando os seus aspetos mais importantes. De seguida, expomos uma breve reflexão sobre o impacto que a tecnologia teve na sociedade e na saúde, decisivo para a resolução de algumas doenças e melhoria da qualidade de vida. Por último, faz-se uma pequena revisão bibliográfica e análise de soluções atualmente existentes na área da toma de medicamentos.

No capítulo 3, *Metodologia, planeamento e análise de requisitos*, apresenta-se a metodologia, o planeamento e a análise de requisitos dos sistemas. É abordada a metodologia de trabalho utilizada, o planeamento realizado para o desenvolvimento do projeto (diagrama de *gantt*) bem como a análise de requisitos da aplicação móvel, descrita através de representação gráfica. Por fim, para o sistema embebido, optou-se por uma planificação através de prototipagem e algoritmia. Através deste método, conseguiu-se uma melhor compreensão e resolução do problema.

No Capítulo 4, Desenvolvimento e implementação *Help Medicine*, é descrito o desenvolvimento e a implementação de ambos sistemas. Descreve-se o processo de desenvolvimento da aplicação móvel, onde serão explicadas as tecnologias e as ferramentas utilizadas, quer para o desenvolvimento da aplicação como para a criação e manutenção da base de dados onde é armazenada toda a informação que diz respeito ao doente e sua respetiva toma de medicação.

Posteriormente, também são descritas todas as etapas inerentes ao desenvolvimento do sistema embebido, começando pela abordagem ao desenvolvimento do *hardware* detalhando as funcionalidades dos componentes utilizados.

O Capítulo 5, *Testes e resultados*, apresenta alguns testes realizados em ambos sistemas bem como os resultados obtidos.

Finalmente, no Capítulo 6, *Conclusões* , são apresentadas as conclusões relativas ao trabalho realizado. São ainda expostas algumas perspectivas de desenvolvimento futuro.

2. As tecnologias e a saúde

Neste capítulo e numa primeira fase, apresenta-se a evolução das tecnologias no mundo, abordando o seu impacto na sociedade e na saúde. Expondo alguma informação que nos permite concluir que felizmente, a tecnologia continua a proporcionar uma verdadeira revolução na sociedade e em particular na área da saúde melhorando a qualidade de vida do ser humano.

Na última secção do presente capítulo são analisados alguns sistemas na área da saúde, que integram Tecnologias da Informação e Comunicação (TIC), de modo a que possamos adquirir um maior conhecimento e perceber alguns requisitos do projeto, explorar ideias e conceitos para o mesmo. As aplicações que foram analisadas dividem-se em dois setores: saúde informativa e medicação.

2.1. A evolução das tecnologias

Desde o ano de 1760 e com a Primeira Revolução Industrial podemos afirmar que as novas tecnologias se incorporaram, e de forma crescente, no quotidiano da vida das pessoas e em toda a sociedade. O desenvolvimento e introdução das tecnologias no nosso dia-a-dia, permitiu uma mudança significativa na sociedade em todos os aspetos e áreas, deixando para traz métodos convencionais que dificultavam a qualidade de vida do ser humano [7].

Há vinte e cinco anos atrás, depositar um cheque ou mesmo levantar dinheiro, significava fazer uma viagem ao banco e interagir com um funcionário. Hoje, os bancos perceberam que não estão apenas a vender um serviço, mas também uma experiência. O banco online, tornou-se uma maneira conveniente dos clientes tratarem das transações no conforto de casa ou em qualquer lugar através dos serviços bancários móveis. Na verdade, muitos utilizadores (cerca de 69% [8]), consideram que foi uma das inovações que melhoraram o estilo de vida.

A tecnologia também teve muitos impactos positivos na vida profissional, agregando conveniência, eficiência e aumentando a produtividade nas empresas que se

tornam cada vez mais digitais. A partilha de arquivos e as tecnologias colaborativas permitem que os funcionários otimizem o fluxo de trabalho e acelerem os processos. O que antes demoraria meses para ficar bem delineado, hoje demora no máximo umas poucas semanas. O cliente já não precisa de se deslocar à empresa para expressar o que pretende no projeto, ferramentas tais como o *email*, *Microsoft Teams*, *Skype* e outras tecnologias colaborativas, vieram dar uma nova forma de comunicação entre empresas e clientes, tornando a necessidade do cliente o foco mais importante no projeto/ferramenta a ser desenvolvido.

O certo é afirmar, que a tecnologia já percorreu um longo caminho, facilitando a vida pessoal e profissional das pessoas e proporcionando um sem fim de oportunidades sem precedentes. À medida que os avanços da tecnologia continuam, as expectativas do ser humano continuarão também a evoluir.

Talvez consideremos, todavia, que a possibilidade de através de uma chamada de voz possamos filtrar as palavras chaves da conversa seja algo do futuro, mas a realidade é bem diferente uma vez que as investigações recentes têm nos mostrado o potencial da inteligência artificial, dos carros sem motoristas e/ou dos robôs domésticos.

Podemos assim afirmar que, a palavra impossível não existe no mundo das TIC. Alias, impossível é uma palavra antiquada para os que trabalham e lidam com a área da tecnologia. Porque devemos dizer que algo é impossível, se há 10 anos atrás o que as pessoas achavam impossível tornou-se hoje realidade? Em pleno século XXI, e como seres humanos, podemos afirmar que as tecnologias são parte integrante das nossas vidas e o que podemos fazer é agradecer e desfrutar desse precioso bem da humanidade.

2.1.1. O impacto da tecnologia na sociedade

A utilização das novas tecnologias permitiu que o dia-a-dia do ser humano sofresse grandes transformações, em particular no modo como nos relacionamos, trabalhamos e como nos comportamos. Vivemos numa era em que as novas tecnologias, determinam novas formas de viver, de trabalhar, de interagir e de sentir. Vejamos o exemplo da educação, onde os quadros interativos já não são novidade em muitas salas

de aulas e graças a estes, é possível criar novas estratégias de ensino e de autoaprendizagem que permitem aos alunos um melhor desempenho [9].

As relações humanas, também foram alteradas pelo uso das tecnologias, sobretudo no que se refere à comunicação interpessoal que tem permitido diminuir distâncias e modos com as pessoas se exprimem e comunicam. Antigamente era muito mais difícil conseguir comunicar com uma pessoa de outro lado do mundo. Hoje, através de dispositivos móveis e/ou de um computador pessoal conseguimos para além de comunicar (voz) também ver a pessoa (vídeo). Isto mostra que o Homem é capaz de se adaptar biologicamente às transformações promovidas pelo rápido desenvolvimento das tecnologias [9].

Para além da vertente pessoal as tecnologias têm tido um forte impacto nas organizações, no modo como estas se relacionam e criam novos negócios. O IFTF (*Institute For The Future*) realizou um estudo em 2017, para mapear o impacto que as novas tecnologias podem ter na vida e no trabalho das pessoas até 2030. O relatório mostra que, na próxima década, todas as organizações e os negócios serão baseados em tecnologia, exigindo que as empresas repensem os modelos atuais de infraestrutura e formas de trabalho. Além disso, o estudo prevê que as tecnologias emergentes – suportadas por enormes avanços em *software*, *big data* e capacidade de processamento – mudarão a forma como as pessoas vivem [10].

Não obstante, é importante que a sociedade estabeleça normas e defina o modo de relacionamento entre o ser humano e máquinas. Outro ponto de destaque do relatório é que, em 2030, a dependência que os humanos têm da tecnologia vai-se transformar numa verdadeira parceria, favorecendo habilidades como a criatividade, o entusiasmo e uma mentalidade empreendedora. Estas habilidades vão proporcionar velocidade, automação e bom desempenho às organizações.

Entre as principais conclusões, deste estudo, destacamos o facto de que o mesmo prevê que, e graças ao avanço tecnológico, aproximadamente 85% das profissões serão novas, ou seja, ainda nem sequer foram inventadas [10].

Podemos assim, afirmar que a tecnologia trouxe grandes benefícios à sociedade, inclusive melhoria na qualidade de vida. O avanço é irreversível e cabe a nós, fazer com que a tecnologia trabalhe a nosso favor, e não contra nós. As tecnologias influenciam o Homem assim como o Homem também influencia o seu desenvolvimento.

Lado a lado será possível criar um mundo melhor, mais sustentável que permita ao ser humano desfrutar de uma vida mais saudável e plena.

2.1.2. O impacto da tecnologia na saúde

A evolução da tecnologia na área saúde ficou mais evidente nos últimos anos. O facto desta área ser uma das mais importantes na sociedade, torna este setor um dos mais preocupados, dinâmicos e rápidos no que toca à utilização e integração de novas tecnologias. A investigação na área da saúde aliada às tecnologias permite que surjam novas técnicas e materiais mais modernos no mercado para a saúde, os quais permitem melhorar a vida dos doentes [11].

Os usos de novas tecnologias num hospital ou clínica, pode passar por um atendimento imediato e um diagnóstico mais preciso especialmente devido ao avanço em áreas como a medicina nuclear. A investigação na área da saúde, permitiu que novos medicamentos e tratamentos médicos possam curar ou controlar múltiplas doenças ajudando as pessoas a viver ativamente e com uma melhor qualidade de vida durante largos anos. Atualmente, contamos com inúmeros métodos que permitem que os médicos examinem os doentes sem o uso de procedimentos invasivos para formar um diagnóstico [12].

Contudo, não é só em aplicações hospitalares que se observa o uso massivo da tecnologia. As aplicações móveis na área da saúde estão a revolucionar como o doente vê e percebe as doenças. Estudos recentes demonstraram que uma grande maioria, cerca de 50% dos doentes [13] usam aplicações móveis para pré diagnosticar doenças, melhorar o bem-estar no dia-a-dia, sugerindo uma série de exercícios, monitorando hábitos de sono e até mesmo proporcionando meditação guiada [14]. Embora não seja uma boa ideia a automedicação ou recorrer a informações da Internet para que o ser humano proceda a um autodiagnóstico, é importante que o doente possa ter conhecimentos e tomar decisões sobre o que fazer perante uma doença.

Do mesmo modo, muitos médicos já perceberam as vantagens do uso dos dispositivos móveis no doente, principalmente idosos e estão a prescrever soluções tecnológicas que permitem entre outros ajudar a monitorizar pontos vitais da saúde, como a pressão arterial, ritmos cardíacos e/ou a gestão da toma de medicação [15]. O uso de

recursos tecnológicos permite ao doente ganhar maior autoconfiança e controlo sobre as suas decisões, permitindo-lhe definir prática de exercício físico, o que comer e quando tomar a medicação.

Em 2012, *Foreman et al.* [16] relataram, num estudo realizado, que os pacientes que adotaram um programa (sistema de alerta) que gera lembretes para a toma da medicação, usando para isso mensagens de texto no telemóvel, mostram uma maior adesão ao tratamento em comparação com indivíduos que não usavam esse sistema de alerta.

Este sistema mostrou ser uma grande ajuda especialmente para idosos com alguma demência, ou então, idosos que vivem sozinhos e não têm qualquer tipo de ajuda para a toma adequada dos respetivos medicamentos. Inevitavelmente, o envelhecimento acarreta grandes alterações a nível cognitivo, o que faz com que a gestão da própria medicação por parte do idoso seja uma problemática. Muitos são os idosos que não aderem a um correto regime de medicação. Muitas das vezes são polimedicados o que os leva a cometerem muitos erros na toma da sua medicação. Tanto o abuso como a falta de um medicamento podem resultar em situações críticas e potencialmente fatais.

Além disso, a evolução na área da saúde também permitiu aos médicos usarem e-mails, textos, vídeos para pesquisa de casos em todo o mundo. Práticas como a telemedicina, são especialmente úteis para médicos e doentes em áreas rurais ou subdesenvolvidas. Sem mover os doentes, os médicos podem consultar especialistas para diagnosticar e tratar, sem a necessidade de acesso a um hospital. Um exemplo em que a telemedicina se utilizou de forma eficaz foi em 2010 após o terremoto no Haiti e, sem dúvida, o seu uso foi importante para salvar inúmeras vidas [17].

2.2. Soluções tecnológicas no âmbito do projeto

Neste subcapítulo são descritas algumas soluções tecnológicas, encontradas no âmbito desta investigação que nos permitem analisar o estado atual das soluções informáticas no âmbito da saúde e em particular na toma da medicação. Numa primeira fase descrevemos soluções que ajudam o utilizador a identificar sintomas fora do

considerado ‘normal’ e por fim soluções que ajudam o utilizador a fazer a gestão da sua medicação.

2.2.1. Soluções sobre saúde informativa

WebMD

A *WebMD*¹ é uma aplicação médica abrangente, que tem como objetivo investigar/detetar sintomas de um modo mais fácil e rápido. Maior parte das vezes em que se pesquisa algo sobre a saúde na Internet, surge uma grande variedade de informação tornando difícil, por vezes, descobrir o que está realmente a acontecer com o doente. Para ajudar a descobrir a patologia do doente, de acordo com a descrição de sintomas, a *WebMD* desenvolveu uma aplicação (Figura 1) que analisa os sintomas (Figura 2), a toma de medicamentos (Figura 3) e identifica possíveis tratamentos de primeiros socorros. Esta aplicação que funciona com recurso da Internet, no entanto permite que o utilizador tenha acesso a informações a algumas funcionalidades primeiros socorros mesmo quando não está conectado [18].

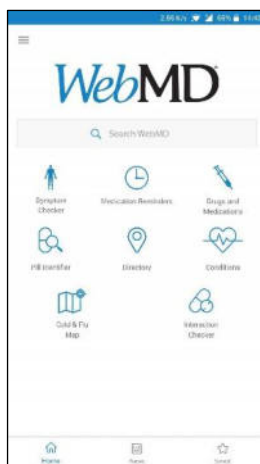


Figura 1 – Ecrã principal da WebMD

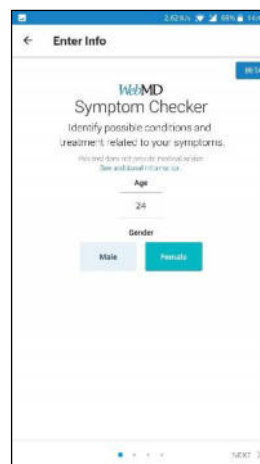


Figura 2 – Ecrã identificar doenças através dos sintomas

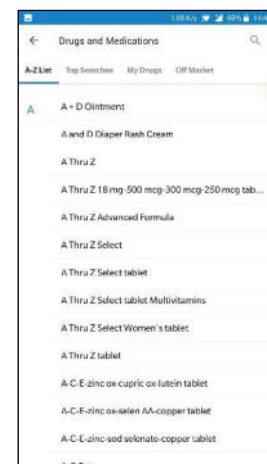


Figura 3 – Ecrã listagem de medicamentos

¹ <https://www.webmd.com/>

Através de testes realizados pela autora da *dissertação*, a aplicação funciona bem e podemos considerar que este é um ótimo complemento para (ou substituir) a *web* na identificação de sintomas e na obtenção de informação. O principal objetivo do *WebMD* é fornecer mais informação sobre um determinado problema, doença, que nos foi diagnosticado, uma vez que os médicos nem sempre dão todas as informações (ou talvez nem sempre ouvimos tudo).

Contudo, a *WebMD* afirma que a aplicação não é um substituto de médico. Ainda assim, os fabricantes recomendam que é conveniente usar esta tecnologia sempre que queiramos obter mais informação sobre os nossos problemas de saúde [18].

MySignals

*MySignals*² é uma plataforma de IoT (*Internet of Things* – Internet das coisas) *eHealth* da empresa *Libelium* (Figura 4), cujo principal objetivo é o de poder de forma imediata identificar um série de problemas de saúde (temperatura, sono, medição de glucose, entre outros). Encontra-se disponível para compra *online* e os preços rondam os 1000 €.



Figura 4 - Sistema *MySignals* [19]

² <http://www.my-signals.com/>

Existem disponíveis no mercado diferentes *kits*, de acordo com as diferentes patologias crônicas que possam existir, todos eles com o mesmo objetivo, o de monitorizar doenças. O *kit* mais completo vem equipado com 15 sensores diferentes que medem os sinais vitais como pulso, taxa de respiração, oxigénio no sangue, temperatura, sinais de eletrocardiograma, pressão arterial, sinais de eletrocardiograma, pressão arterial, sinais eletromiográficos musculares, níveis de glicose, pele galvânica, capacidade pulmonar, ressonar ondas, posição do doente, fluxo aéreo e parâmetros da escala corporal. A vigilância remota do *MySignals* permite que os utilizadores monitorizem o seu estado de saúde diariamente, de acordo com o que médico prescreveu [20].

Além disso, a plataforma (Figura 5) permite a visualização em tempo real dos dados recolhidos, que estão a ser armazenados na *cloud* do *MySignals*. Também é possível gerir diferentes perfis de utilizadores [19].



Figura 5 – Cloud do *MySignals* [19]

2.2.2. Soluções sobre medicação

Medisafe Medication Reminder

O *MediSafe*³ é uma aplicação disponível gratuitamente para *Android* e *iOS* que aborda o problema da toma de medicamentos a tempo e horas (Figura 6). A empresa afirma - com base em seus dados internos - que quando os doentes começam a usar a

³ <https://www.medisafe.com/>

aplicação, a toma a tempo e horas aumenta para uma média de 86%, em comparação com 50% dos doentes que não utilizam esta tecnologia [21].

A aplicação pode ser usada pela família e/ou cuidadores. É possível sincronizar dispositivos com a aplicação e alertar deste modo se uma pessoa não toma sua medicação ou se decide ignorar a toma. O *MediSafe* é especialmente importante para os cuidadores que fazem a gestão de vários membros da família.

A interface do utilizador é um dos pontos fortes do *MediSafe* (Figura 7). Depois de testada podemos afirmar que a mesma é muito fácil de usar e flexível com os dados. Os ícones são personalizáveis e os gráficos mostram, de modo perceptível, com que frequência é realizada a toma de medicação.

A interface foi projetada para funcionar muito bem com o *Android*, com um *widget* da janela inicial (Figura 8), notificações de som, notificações de LED, configurações de vibração e um *feed* de notícias (Figura 9) que é composto por vídeos e dicas sobre como gerir a condição do doente.

Ainda que a aplicação tenha sido desenvolvida para trabalhar on-line é também possível utilizar a mesma parcialmente em modo *offline*. A toma de medicamentos e o uso de gráficos são as principais funcionalidades quando não há conectividade.

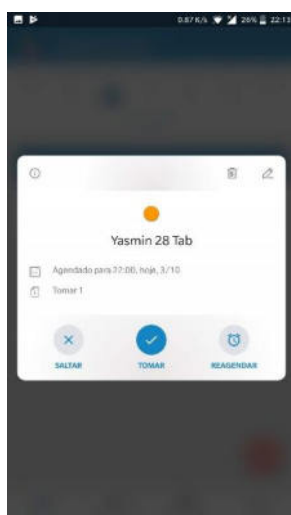


Figura 6 – Visualização do alerta do medicamento da aplicação *MediSafe*

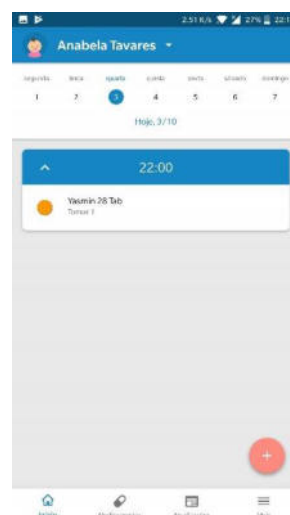


Figura 7 – Ecrã principal da aplicação *MediSafe*

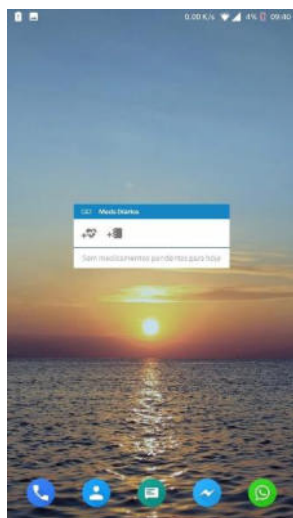


Figura 8 – Widget da aplicação Medisafe

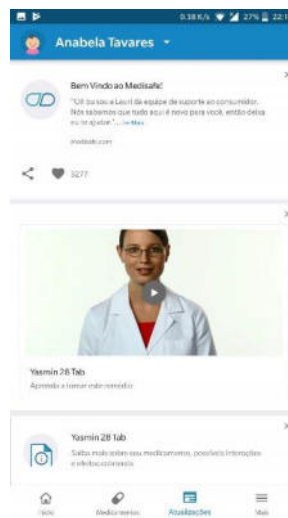


Figura 9 - Feed de notícias da aplicação Medisafe

Maya e Jon

A empresa MedMinder⁴, é responsável pelo avanço da tecnologia na área da saúde, pela construção de diversos dispensadores que têm como objetivo o de ajudar na toma de medicação. Na análise realizada, optou-se por estudar dois dos dispensadores projetados por esta empresa: o Maya (Figura 10) e o Jon (Figura 11).

Em qualquer um destes dispensadores os comprimidos são organizados em compartimentos numa matriz 4x7, sendo os comprimidos colocados em pequenos recipientes. Nenhum dos dispensadores possuem ecrã LCD (*Liquid crystal display*) nem botões e toda a sua parametrização é realizada através de uma página web. Os dispositivos, possuem um modem GSM (*Global System for Mobile Communications*) para enviar os registos das anomalias e receber a parametrização consoante prescrição médica do doente.

A diferença entre os dois modelos é o facto de o dispensador *Jon* ter todos os compartimentos bloqueados, impossibilitando o acesso indevido às tomas. A supervisão pode ser partilhada entre o médico e os familiares ou cuidadores. A empresa também disponibiliza um serviço de entrega dos dispensadores já carregados com a prescrição médica do paciente, simplificando assim a tarefa do cuidador [22].

⁴ <https://www.medminder.com/>



Figura 10 – Dispensador MedMinder – Maya
[23]



Figura 11 - Dispensador MedMinder – Jon [23]

Após apresentação, revisão bibliográfica e testes de algumas aplicações móveis e dispensadores automáticos de medicamentos à venda no mercado, foram aproveitadas algumas funcionalidades que estes demonstravam. Para a aplicação móvel aproveitou-se a ideia do calendário geral com os alertas que existiam, o menu de abas para apresentação do conteúdo da aplicação e o *fab button* para uma navegação mais rápida. Já para o dispositivo eletrónico o formato do protótipo e *led* para alerta luminoso foi algo que chamou à atenção.

3. Metodologia, planeamento e análise de requisitos

Tal como descrito na parte introdutória o presente capítulo descreve o planeamento, bem como as etapas de desenvolvimento que foram realizados para o desenvolvimento dos dois sistemas propostos (software e hardware), que permitirão um controlo da toma de medicação. Nesta secção é apresentada a metodologia utilizada no desenvolvimento do projeto, bem como uma breve explicação do porquê da escolha da metodologia *XP* (*eXtreme Programming*). Seguidamente, também expomos o processo de planeamento do projeto realizado.

Posteriormente, é descrito o processo da análise dos requisitos onde são abordadas as técnicas de levantamento de requisitos utilizadas, assim como os requisitos funcionais e não funcionais dos sistemas.

Por último e ainda no presente capítulo, apresenta-se a planificação do sistema embebido realizada através de prototipagem e de algoritmia.

3.1. Metodologia

O planeamento e desenvolvimento deste projeto esteve sujeito a algumas mudanças ao longo do seu desenvolvimento e as tarefas foram divididas em pequenas partes simples que no fim compuseram duas soluções complexas. Ao longo do processo de desenvolvimento do projeto foi necessário ter comunicação com o orientador do projeto para retirar dúvidas ou até mesmo especificar novos requisitos. Tendo por base estes pressupostos a metodologia de desenvolvimento de software que foi aplicada neste projeto é a Programação Extrema.

Esta metodologia consiste em testar constantemente a aplicação de forma a que esta cumpra com todos os requisitos estipulados e tendo em vista a menor probabilidade de erro. A cada interação foram adicionadas novas funções à aplicação de

forma progressiva sendo que cada interação, revisão, esteve sujeita a alterações, de modo a cumprir com o objetivo inicial.

O desenvolvimento do projeto assentou nas seguintes fases:

1. Análise de Requisitos (Análise de requisitos da aplicação móvel) – Discussão e análise com o orientador acerca dos requisitos que o sistema teria que ter.
2. Implementação/Programação (Desenvolvimento e implementação *Help Medicine*) – Nesta fase foi desenvolvido o código dos sistemas, isto é, implementados os requisitos do sistema.
3. Testes (Testes e resultados) – Foram realizados ao longo de todas as fases do projeto testes (requisitos, implementação) de forma a respeitar todos os requisitos de software estipulados e, assim, atingir os objetivos do mesmo com qualidade.
4. Documentação – Foi descrito detalhadamente o processo de planeamento e desenvolvimento.

A especificação detalhada de cada uma das fases encontra-se descrita nas seções seguintes.

3.2. Planeamento

A gestão e o desenvolvimento de um projeto de *software* dependem de um planeamento eficaz e eficiente logo aquando do início da definição do mesmo. Um projeto inicial bem planeado resolve metade dos problemas que possam ser definidos inicialmente [24].

O processo de planeamento do sistema *Help Medicine* iniciou-se com a análise dos requisitos que tinham sido previamente identificados. Esta análise foi realizada tendo em consideração e em conjunto com a pesquisa realizada sobre outras aplicações existentes no mercado. Esta análise permitiu-nos definir a estrutura e as funcionalidades que os sistemas deveriam ter.

Logo após a fase de desenho e modelação, iniciámos o desenvolvimento dos sistemas. Em simultâneo foram realizados testes que permitiram que o projeto fosse corrigido sempre que houvesse necessidade, garantido assim que o sistema ficasse preparado para o utilizador final mais rapidamente e com qualidade.

O gráfico da Figura 12 ilustra as principais etapas e a percentagem de tempo gasta em cada fase do projeto. Os resultados mostram que o tempo gasto nas várias etapas não difere muito, porque como já foi referido anteriormente, foram realizadas tarefas em simultâneo.

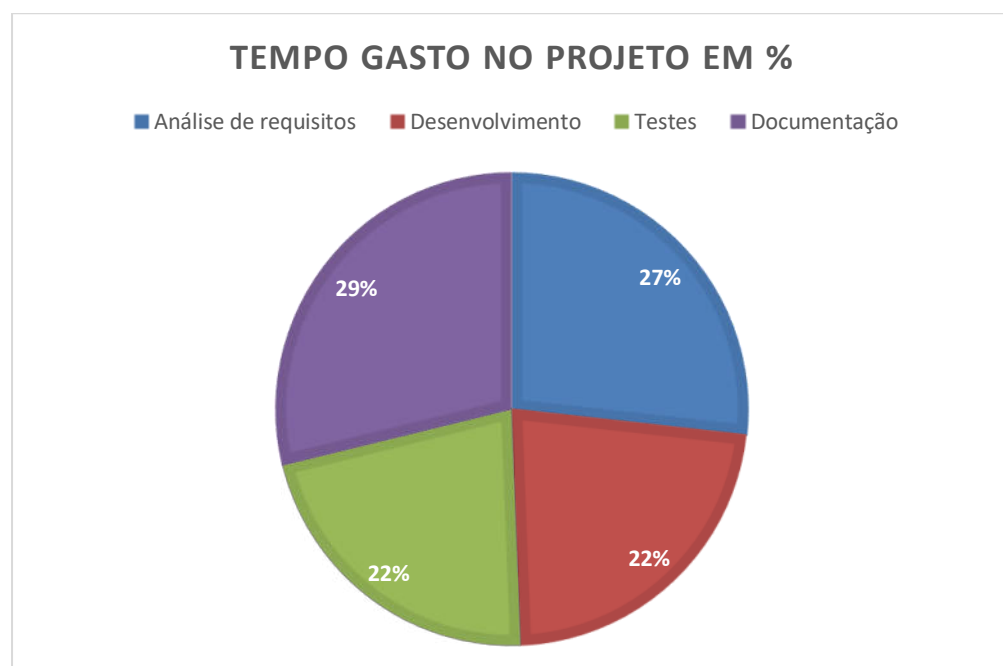


Figura 12 - Percentagem de tempo gasto por atividade

O plano de trabalho correspondente a cada etapa encontra-se definido no diagrama de *Gantt* (Figura 13). Ao analisarmos o diagrama, verificamos que o comprimento das barras está associado a um determinado tempo para a conclusão de uma etapa, podendo concluir que a documentação foi a que demorou mais tempo a realizar.

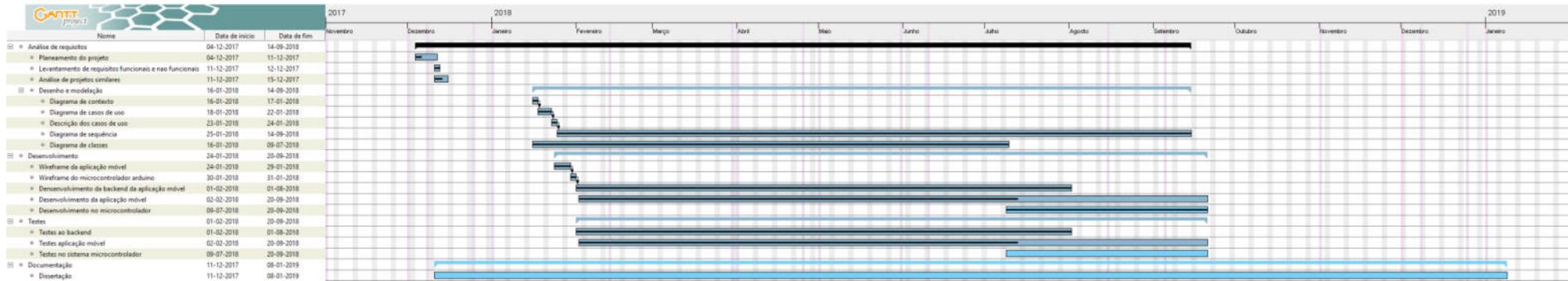


Figura 13 – Etapas do planeamento apresentadas através do diagrama de *Gantt*

3.3. Análise de requisitos da aplicação móvel

Depois do planeamento, o próximo passo é o levantamento e a análise de requisitos. A análise de requisitos é considerada uma das principais atividades no processo de levantamento de requisitos [25]. É também uma das etapas vitais estipuladas para o desenvolvimento de um software, independentemente do processo escolhido [26].

Esta fase tem como objetivo analisar e apresentar todos os requisitos necessários para o desenvolvimento do software de forma a que o cliente/utilizador final compreenda e confirme se todas as funcionalidades estão contempladas [27]. Segundo sugere *Pressman* [28], é através da análise de requisitos que se consegue associar todos os requisitos com a fase do desenho de software.

Nesta secção, são abordadas e descritas as técnicas de levantamento utilizadas. Seguidamente, apresentamos os requisitos funcionais e não funcionais dos sistemas e por último, o desenho e a modelação necessária para a concretização do projeto em si.

3.3.1. Técnicas de recolha e levantamento de requisitos

O levantamento de requisitos é uma parte fundamental no desenvolvimento de um sistema e também a mais complexa. O mais importante no levantamento de requisitos é entender aquilo que o cliente deseja ou o que o cliente acredita que precisa no projeto [29].

Para ajudar neste processo, utiliza-se técnicas de recolha de dados que permitem uma melhor compreensão dos requisitos. Podemos afirmar que não existe uma técnica que seja melhor ou pior que outra, estas devem ser adequadas consoante o projeto. Também não é obrigatório só usar uma, dependendo da situação é possível utilizar várias abordagens para chegar ao melhor resultado [30].

Nos pontos seguintes, abordam-se as diferentes técnicas utilizadas no projeto que permitiram o levantamento e a análise de requisitos.

3.3.1.1. Análise de documentos

A análise de documentos passa pelo estudo de documentação disponível sobre uma ou várias soluções existentes, para a identificação de informações relevantes que nos permitem posteriormente desenvolver uma nova solução. A principal vantagem desta técnica (análise de documentos) é que não se trata de um trabalho iniciado do zero, uma vez que se aproveita/analisa material existente para descobrir ou validar requisitos de um sistema [31].

Com o objetivo de recolher dados pertinentes, já que não existia um cliente/utilizador para os definir, foi utilizada esta técnica no planeamento do projeto. Foram analisados vários documentos e também testadas as aplicações que se enquadravam no tema do projeto que já existiam no mercado, como foi especificado no capítulo 2. Através desta análise conseguiu-se identificar alguns requisitos que permitiram identificar funcionalidades que dão valor à aplicação que pretendemos desenvolver.

À medida que a pesquisa ia avançando, foi necessário analisar outros documentos, tais como revisões (*reviews*) de especialistas na área da saúde, para complementar a análise previamente realizada.

3.3.1.2. Prototipagem

A prototipagem é um processo que tem como objetivo facilitar o entendimento dos requisitos, apresentar conceitos e funcionalidades do sistema. Desta forma, podemos propor uma solução adequada para o problema, aumentando a perceção do cliente/utilizador [31].

Estes podem ser desenvolvidos em diferentes níveis de fidelidade: quanto maior for a fidelidade, mais o protótipo se assemelhará ao resultado entregue. No entanto, um protótipo de alta fidelidade leva mais tempo para ser criado ou modificado. A escolha do protótipo ideal varia de acordo com o nível de entendimento do negócio, a complexidade dos requisitos, prazo e orçamento para elaboração [31].

Podemos dividir os protótipos em três categorias:

1. Wireframes

São protótipos de baixa fidelidade, rápidos para se desenvolver e modificar. Os *Wireframes* não vão mostrar detalhes visuais ou interações do ecrã, mas vão ajudar a validar os requisitos de maneira eficiente. São a melhor escolha para representar cenários complexos onde um fluxo ou processo precisa ser compreendido [31], [32].

2. Protótipos visuais

Criados com programas de edição gráfica, estes protótipos têm maior apelo visual, entretanto, não possuem interações de ecrã e pedem mais tempo para fazer ajustes e melhorias. São uma ótima opção para ecrãs com maior ênfase em estética e usabilidade, quando os requisitos já foram entendidos [31], [32].

3. Protótipos interativos

São protótipos completos e representativos. Além da parte visual, englobam uma série de detalhes de estética, proporcionando uma experiência rica e realista. Também ajudam a equipa a identificar novos requisitos, oportunidades e futuros problemas [31], [32].

No planeamento do projeto desta dissertação, optou-se por escolher esta técnica para visualizar melhor o produto final (capítulo 3, ponto 3.4). Com a construção de *wireframes* (capítulo 3, ponto 3.3.3.5) e protótipos visuais conseguiu-se identificar mais requisitos e funcionalidades que ainda não tinham sido identificados com a análise de documentos.

3.3.2. Requisitos funcionais e não funcionais

A construção de um sistema, na maioria das vezes, é a união das etapas de planeamento e desenvolvimento. Compreender como o sistema vai funcionar e quais as suas funções é um trabalho que tem de ser feito com precisão. Como tal, a definição dos requisitos deve ser simples e objetiva a fim de facilitar o desenvolvimento.

Os requisitos funcionais do sistema, descrevem as funcionalidades que se espera que o sistema forneça. Estes dependem do sistema que está a ser desenvolvido e

quais são os utilizadores que o vão utilizar. De modo bastante geral, os requisitos funcionais descrevem a função do sistema detalhadamente, as suas entradas e saídas, exceções, entre outros [33].

Os requisitos não funcionais, como o nome sugere, são aqueles que não dizem respeito diretamente às funções específicas do sistema. Estes podem estar relacionados as propriedades de sistema, como confiabilidade, tempo de resposta e espaço em disco. Como alternativa, eles podem definir restrições para o sistema, como a capacidade dos dispositivos de E/S (entrada/saída) e representações dos dados utilizados nas interfaces do sistema [33].

Os requisitos podem ser expressos de diversas maneiras. Normalmente são apresentados em forma textual, tabelas ou então em diagramas. No desenvolvimento dos sistemas optou-se por apresentar os requisitos em tabelas. De acordo com cada um dos sistemas apresentam-se os respetivos requisitos identificados.

A aplicação móvel *Help Medicine* deve respeitar, pelo menos, os seguintes requisitos funcionais (Tabela 1) e não funcionais (Tabela 2):

Tabela 1 – Requisitos Funcionais da aplicação *Help Medicine*.

ID do Requisito	Requisito
RF01	O sistema deve permitir que o utilizador efetue o registo.
RF02	O sistema deve permitir que o utilizador consulte os seus dados.
RF03	O sistema deve permitir que o utilizador altere os seus dados.
RF04	O sistema deve permitir que o utilizador registe alertas.
RF05	O sistema deve permitir que o utilizador altere alertas.
RF06	O sistema deve permitir que o utilizador exclua alertas.
RF07	O sistema deve permitir que o utilizador consulte alertas.
RF08	O sistema deve permitir que o utilizador registe medicamentos.
RF09	O sistema deve permitir que o utilizador consulte os medicamentos.
RF10	O sistema deve permitir que o utilizador registe encomendas.
RF11	O sistema deve permitir que o utilizador consulte encomendas.
RF12	O sistema deve permitir que o utilizador registe farmácias.
RF13	O sistema deve permitir que o utilizador consulte farmácias.

RF14	O sistema deve permitir que o utilizador receba notificações quando estiver na hora de tomar a medicação (mesmo desligado).
------	---

Tabela 2 – Requisitos não funcionais da aplicação *Help Medicine*

ID do Requisito	Requisitos
RNF01	O tempo de resposta da aplicação não deve ultrapassar 30 segundos.
RNF02	O sistema deve garantir que os dados estão protegidos de acesso não autorizados.
RNF03	A aplicação deve ser compatível com versões anteriores ao android 6.0.
RNF054	O sistema deve ser implementado em <i>NodeJS</i> com base de dados <i>MySQL</i> .

No que diz respeito ao sistema embebido, este deve compreender os seguintes requisitos funcionais (Tabela 3) e não funcionais (Tabela 4):

Tabela 3 – Requisitos funcionais do sistema embebido

ID do Requisito	Requisito
RF01	O sistema deve permitir armazenar pelos menos 21 tomas.
RF02	O sistema deve permitir que o utilizador seja alertado de tomar a medicação.
RF04	O sistema deve permitir o utilizador definir a hora e data da toma de medicação.
RF05	O sistema deve permitir que o utilizador edite alertas.
RF06	O sistema deve permitir que o utilizador apague alertas

Tabela 4 – Requisitos não funcionais do sistema embebido.

ID do Requisito	Requisitos
-----------------	------------

RNF01	O sistema deve ser extensível, permitindo que no futuro sejam colocadas novas funcionalidades.
RNF02	O sistema deve ser facilmente entendido.
RNF03	O sistema deve ter um conjunto de imagens distintas associadas a cada tipo de alerta.
RNF04	O sistema deve ter um manual do utilizador que explique como usar o sistema.

3.3.3. Desenho e modelação

Sendo um projeto realizado individualmente, isto é não existe a integração uma equipa, houve a necessidade de o próprio ser analista, gestor de projeto, programador dos sistemas, *tester* e utilizador. Deste modo, e para melhor compreensão dos requisitos do projeto foi essencial analisar potenciais erros e riscos através da modelação de *software*.

A modelação utiliza a representação gráfica, através de distintos modelos, para descrever o problema a ser resolvido, o que permitiu ter uma melhor visão de como a aplicação irá funcionar e como esta deverá ser implementada.

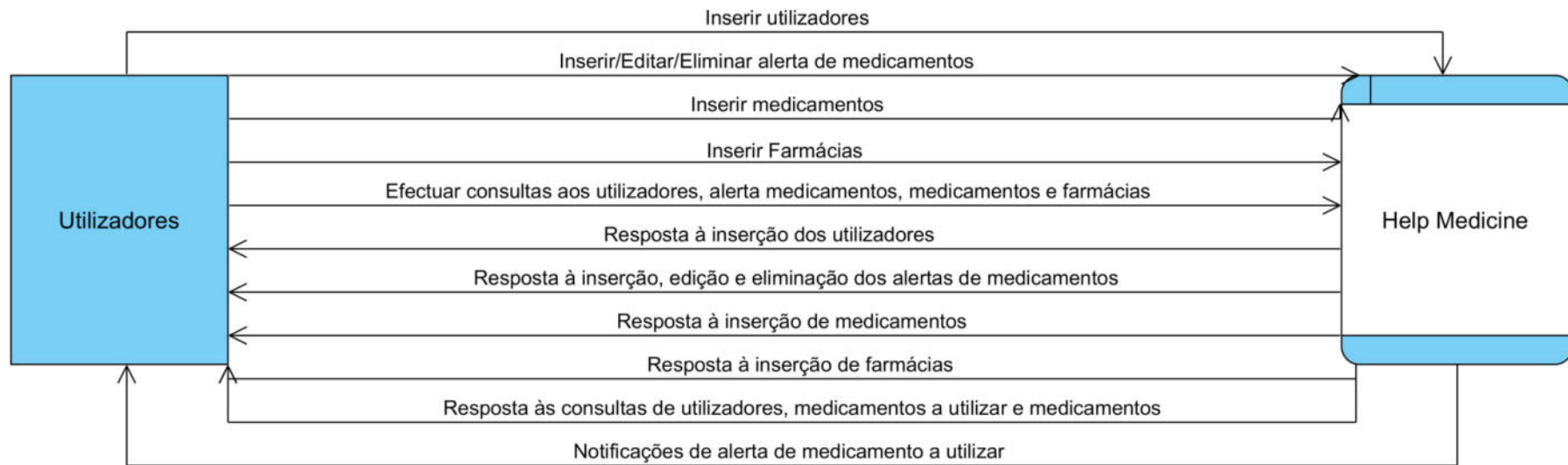
Neste subcapítulo, é apresentado o diagrama de contexto e os principais diagrama de *software* utilizando a notação UML (*Unified Modelling Language*). Também será apresentado o *wireframe* realizado para identificação de mais algumas funcionalidades e requisitos da aplicação. Além disso, também será dada uma breve explicação de cada um deles para um melhor enquadramento do sistema. Os restantes diagramas de *software* e *wireframes* encontram-se no Anexo A – Análise de requisitos Anexo A – Análise de requisitos Anexo A – Análise de requisitos Anexo B – *Wireframes*.

3.3.3.1. Diagrama de contexto

O diagrama de contexto ou DFD nível 0, mostra de forma intuitiva como funciona o sistema. Com este diagrama, podemos ver o fluxo de dados entre as entidades

externas e o sistema. Desta forma, este diagrama serve para validação dos requisitos do projeto, assim como, também poderá ser útil em discussões futuras sobre o projeto [33].

A Figura 14, ilustra o diagrama de contexto da aplicação *Help Medicine*. Este enumera os requisitos enunciados no ponto 3.3.2 (Requisitos funcionais e não funcionais), onde identifica o fluxo de dados e a direção da comunicação entre a aplicação e o ator (Utilizador).

Figura 14 - Diagrama de contexto da aplicação *Help Medicine*

3.3.3.2. Diagrama de casos de uso

Os diagramas de casos de uso, passam por identificar os principais momentos de interação dos utilizadores com o sistema, de maneira a que se obtenha os passos necessário para atingir um objetivo específico. Também são utilizados para assegurar que tanto o utilizador final como o programador, possuem um entendimento comum dos requisitos. O objetivo é mostrar o que um sistema deve possuir e não como o vai fazer [28] e [33].

Na Figura 15, apresenta-se o diagrama de casos de uso da aplicação *Help Medicine*.

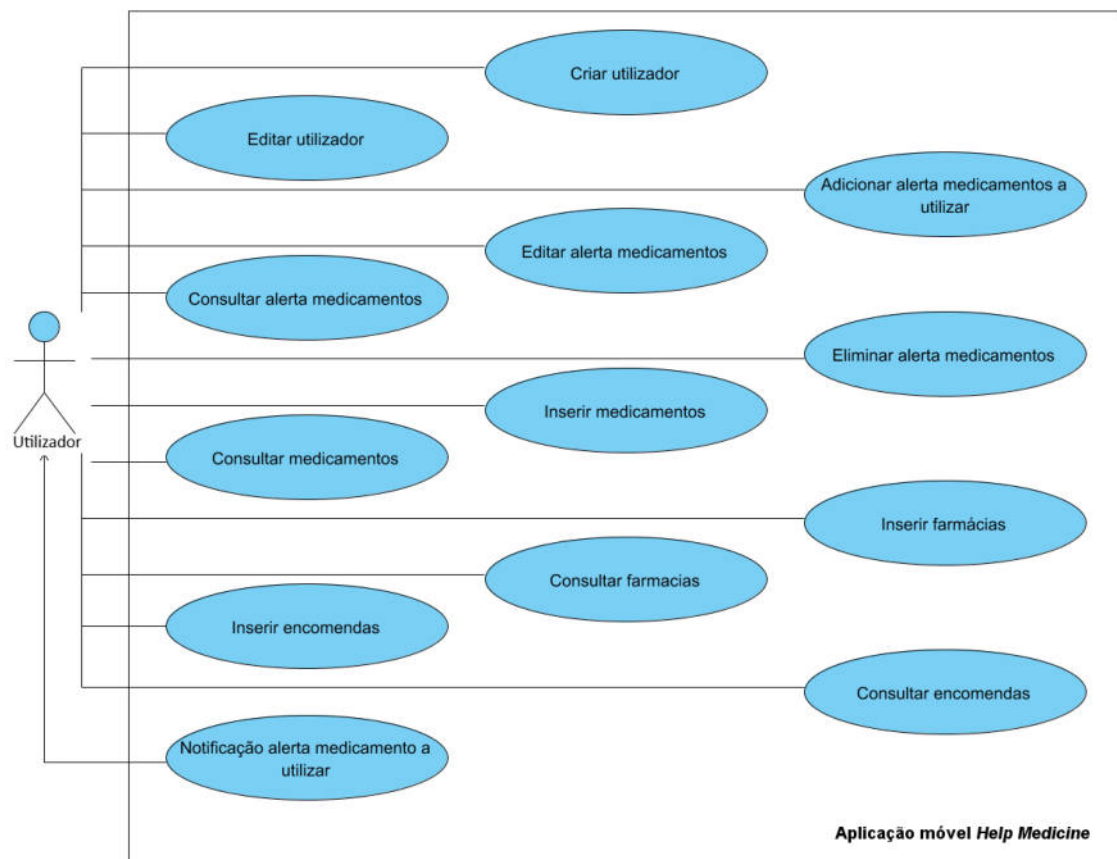


Figura 15 – Diagrama de casos de uso da aplicação *Help Medicine*

A aplicação terá apenas um ator, denominado “Utilizador”, que representa o quem irá usar a aplicação. O utilizador terá acesso às várias funções da aplicação, como por exemplo inserir doente, editar doente, entre outras.

No entanto, relembramos que o ponto fundamental da aplicação é a gestão dos alertas da toma de medicamentos. Com este objetivo, o utilizador do sistema poderá para além de criar os seus próprios alarmes, ver todos os alarmes registados, podendo para cada um deles remover ou editar os mesmos. A edição do alarme consistirá essencialmente em mudar a hora, dia e se pretende ativar ou desativar o alerta do medicamento.

No caso de pretender adicionar um novo alarme, o utilizador terá a possibilidade de escolher o medicamento através de uma *auto-complete list*, adicionar a hora da toma, a data de início do alarme, a data de fim e também adicionar a data de validade do mesmo. O passo que concluirá o processo de criação de alertas será feito pela aplicação. A aplicação irá agendar as notificações locais no momento da finalização da criação do alerta. A partir daqui o sistema do dispositivo assume a responsabilidade de entregar a notificação ao utilizador no momento de tomar a medicação.

3.3.3.3. Descrição de casos de uso e Diagramas de sequência

A descrição de um caso de uso é uma representação narrativa do comportamento do sistema. Esta descrição ajuda os analistas a identificar as tabelas no sistema e a compreender o que o sistema deve fazer. É utilizada uma forma exata dessa descrição para identificar quais as informações trocadas, como a interação é iniciada, entre outros [33].

É possível utilizar qualquer técnica para descrever os casos de uso desde que a descrição seja breve e facilmente compreensível. No entanto, um conjunto mínimo de elementos deve estar presente na maioria das descrições [28]. Estes são:

- **Atores:** descrição dos atores envolvidos no caso de uso;
- **Pré-condições:** regras de estado que definem como o sistema deve se encontrar antes de entrar no caso de uso;
- **Pós-condições:** regras de estado que definem como o sistema deve se encontrar depois de executado o caso de uso.

- **Caminho principal:** atividades que ocorrem entre os atores e o sistema à medida que interagem para atingir um objetivo;

Durante a descrição do caso de uso podem ocorrer exceções, o que implica criar um caminho alternativo que pode retornar ao caminho principal que leva ao objetivo final ou então terminar sem sucesso [28].

De forma a complementar as descrições dos casos de uso, serão apresentados simultaneamente os diagramas de sequência. Um dos modelos mais dinâmicos, úteis e fáceis de compreender. Estes documentam, para cada interação, a sequência de eventos que acontecem entre interfaces e os objetos (tabelas da base de dados) [28], [33].

Num diagrama de sequência [28]:

1. As interfaces/tabelas envolvidas na interação são organizadas horizontalmente, com uma linha vertical.
2. O tempo é representado verticalmente, de modo a que progrida para baixo nas linhas verticais. Portanto a sequência de operações pode ser lida facilmente.
3. As interações entre objetos são representadas por setas rotuladas que ligam as linhas verticais. Estas não são o fluxo de dados, mas representam mensagens ou eventos que são fundamentais para a interação.
4. O retângulo estreito, na linha vertical, representa o tempo pelo qual a interface/tabela está no controlo do sistema. Uma interface/tabela assume o controlo na parte superior do retângulo e transfere mensagens para outra interface. Se houver uma hierarquia de chamadas, o controlo não será cedido até que retorne à chamada inicial.

Apresentam-se de seguida e de forma mais detalhada a descrição dos casos de uso e diagramas de sequência mais importantes do sistema, sendo que os restantes diagramas que complementa a análise e modelação se encontram em anexo.

3.3.3.3.1. Descrição do caso de uso “Criar utilizador”

Tabela 5 – Descrição do caso de uso “Criar utilizador”

Nome	Criar utilizador
Objetivo	Este caso de uso tem como objetivo principal adicionar um utilizador à aplicação.
Atores	Utilizador
Pré-condição	O dispositivo móvel estar identificado e registado.
Pós-condição	Registar utilizador no sistema.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a opção “Criar novo utilizador”. 2. O sistema exibe o formulário para criar um novo utilizador. 3. O ator seleciona a opção “Escolhe foto”. 4. O sistema atualiza a foto no formulário. 5. O ator introduz os dados restantes (nome, idade, médica de família, problemas de saúde). 6. O ator clica no botão guardar para confirmar e enviar os dados. 7. O sistema guarda o novo utilizador. 8. O sistema encaminha o utilizador para a página inicial e retorna mensagem de sucesso.
Caminho alternativo	<ol style="list-style-type: none"> 3. O sistema exibe as opções, “Tirar foto” ou “Escolher foto da Galeria”. <ol style="list-style-type: none"> a) Se o ator optar por “Tirar foto” <ol style="list-style-type: none"> 3.1.a.1. O sistema abre a câmara do dispositivo; b) Se o ator optar por “Escolher foto da Galeria” <ol style="list-style-type: none"> 3.1.b.1. O sistema abre a galeria do dispositivo; c) Voltar para o ponto 4 do caminho principal.

Caminho de exceção	3. a) O sistema pode não mostrar as opções para a foto, não foram ativadas as permissões. 6. a) O ator não preencheu os dados corretamente, retorna para o ponto 5 do caminho principal, sinalizando quais os campos que não foram preenchidos de forma correta.
Suplementos ou adornos (requisitos segurança)	Fazer vários testes um de verifique: a) Se o utilizador é capaz de tirar foto; b) Se o utilizador é capaz de escolher uma foto da galeria; c) Se o utilizador é capaz de inserir todos os dados;

A Figura 16 mostra o diagrama de sequência do caso de uso “Criar utilizador” em que as interfaces “Criar utilizador”, “Gestor dos dados utilizador” e a tabela utilizador da base de dados estão no topo do diagrama.

Neste cenário, a interface “Criar utilizador” recebe um pedido do “Utilizador” da aplicação para enviar o formulário de registo do utilizador. A interface responde apresentando o formulário. O “Utilizador” preenche os dados do utilizador e submete-os. A interface “Criar utilizador” e envia os dados para a interface seguinte “Gestor dos dados utilizador”. Esta posteriormente confirma os dados recebidos e verifica se existe algum problema com a sua introdução/validação. Se não for identificado nenhum problema, a interface redireciona os dados para a base de dados (Tabela utilizador). Posteriormente, a interface regista os dados retornando uma mensagem de sucesso para o “Utilizador”. Caso se identifique algum problema de dados a interface “Gestor dos dados utilizador” não permite a inserção de dados na base de dados e emite uma mensagem ao “Utilizador” para corrigir os mesmos.

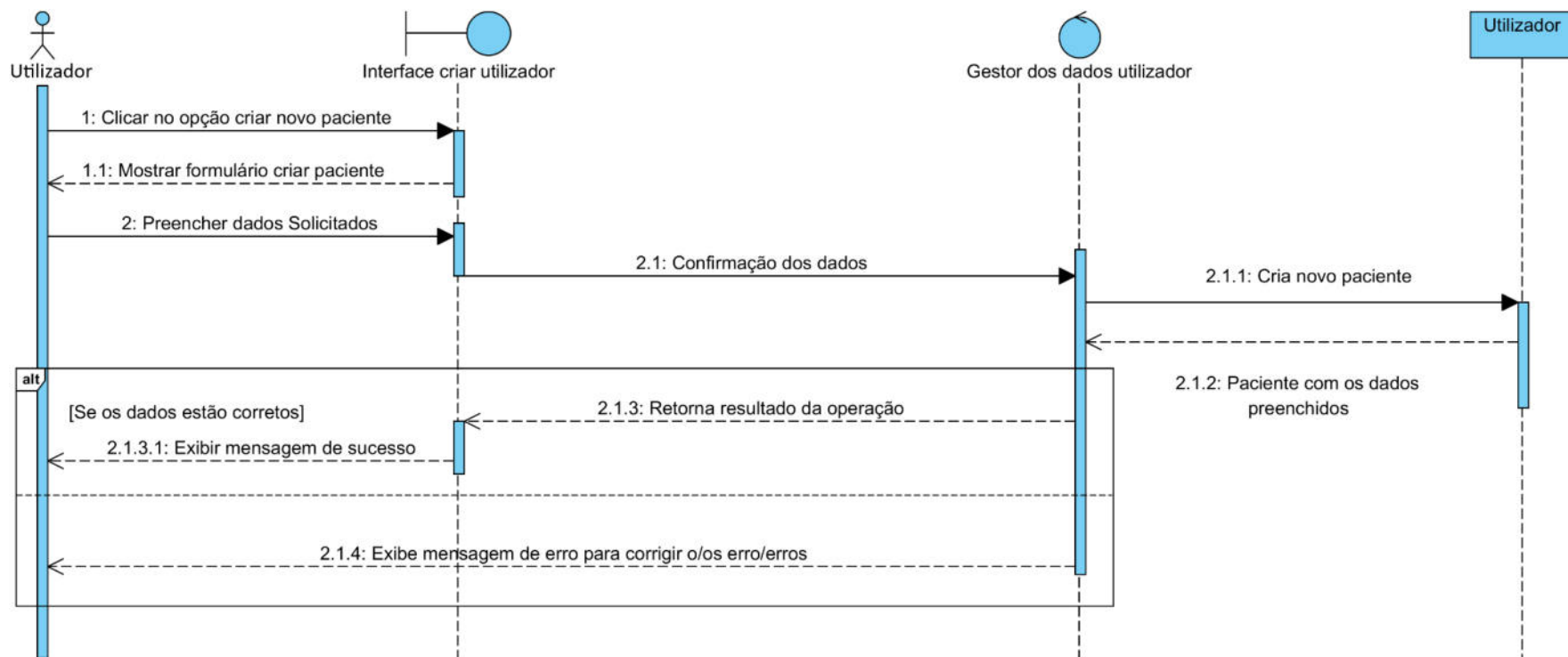


Figura 16 – Sequência de operações – “Criar Utilizador”

3.3.3.3.2. Descrição do caso de uso “Adicionar alerta medicamentos”

Tabela 6 - Descrição do caso de uso “Adicionar alerta medicamentos”

Nome	Adicionar alerta medicamentos
Objetivo	Este caso de uso tem como objetivo selecionar qual o medicamento que o utilizador pretende ser alertado.
Atores	Utilizador
Pré-condição	Lista de medicamentos já preenchida; Utilizador no seu perfil.
Pós-condição	Registar novo alerta no sistema.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a opção “Adicionar alerta”. 2. O sistema exibe o formulário para a criação do alerta. 3. O ator preenche o nome do medicamento. <ol style="list-style-type: none"> 3.1. O sistema executa o caso de uso “Consulta de medicamentos”, de acordo com o que o ator digita na caixa de texto. 4. O ator preenche os restantes dados (hora, data de início, data de fim, dosagem e data de validade). 5. O ator clica no botão guardar para confirmar e inserir os dados. 6. O sistema guarda o novo alerta. 7. O sistema encaminha o utilizador para o ecrã dos alertas e retorna mensagem de sucesso do novo alerta.
Caminhos de exceção	<ol style="list-style-type: none"> 3. a) o sistema não executa o caso de uso “Consulta de medicamentos”. 5. a) O ator não preencheu os dados corretamente, retorna para o ponto 4 do caminho principal, sinalizando quais os campos que não foram preenchidos de forma correta.

Suplementos ou adornos (requisitos segurança)	Fazer vários testes um de verifique: a) Se o utilizador consegue ver a lista de medicamentos através da caixa de texto.
--	--

A Figura 17 apresenta o diagrama de sequência do caso de uso “Adicionar alerta de medicamentos” em que as interfaces “adicionar alerta”, “Gestor dados alerta” e as tabelas medicamentos e alerta de medicamentos da base de dados estão no do topo do diagrama.

Neste cenário, a interface “adicionar alerta” recebe um pedido do “Utilizador” da aplicação para enviar o formulário de registo do alerta. A interface responde apresentando o respetivo formulário.

Posteriormente o “Utilizador” preenche os dados, começando por identificar/pesquisar o medicamento que deverá tomar. A interface envia o pedido de consulta à base de dados que retorna com os dados do medicamento. Em seguida, o “Utilizador” preenche os restantes dados relativos à toma de medicamento e submete o alerta no sistema.

A interface “Gestor dados de alerta” confirma os dados recebidos e verifica se há algum problema. Se não houver nenhum problema, de introdução e validação de dados, a interface redireciona os dados para a base de dados (Tabela alerta de medicamentos). Posteriormente, é retornada uma mensagem de sucesso para o “Utilizador”. Caso exista algum problema, a interface “Gestor dos dados alerta” não deixa prosseguir o registo de toma de medicamento, na base de dados, e emite uma mensagem ao “Utilizador” para corrigir os dados.

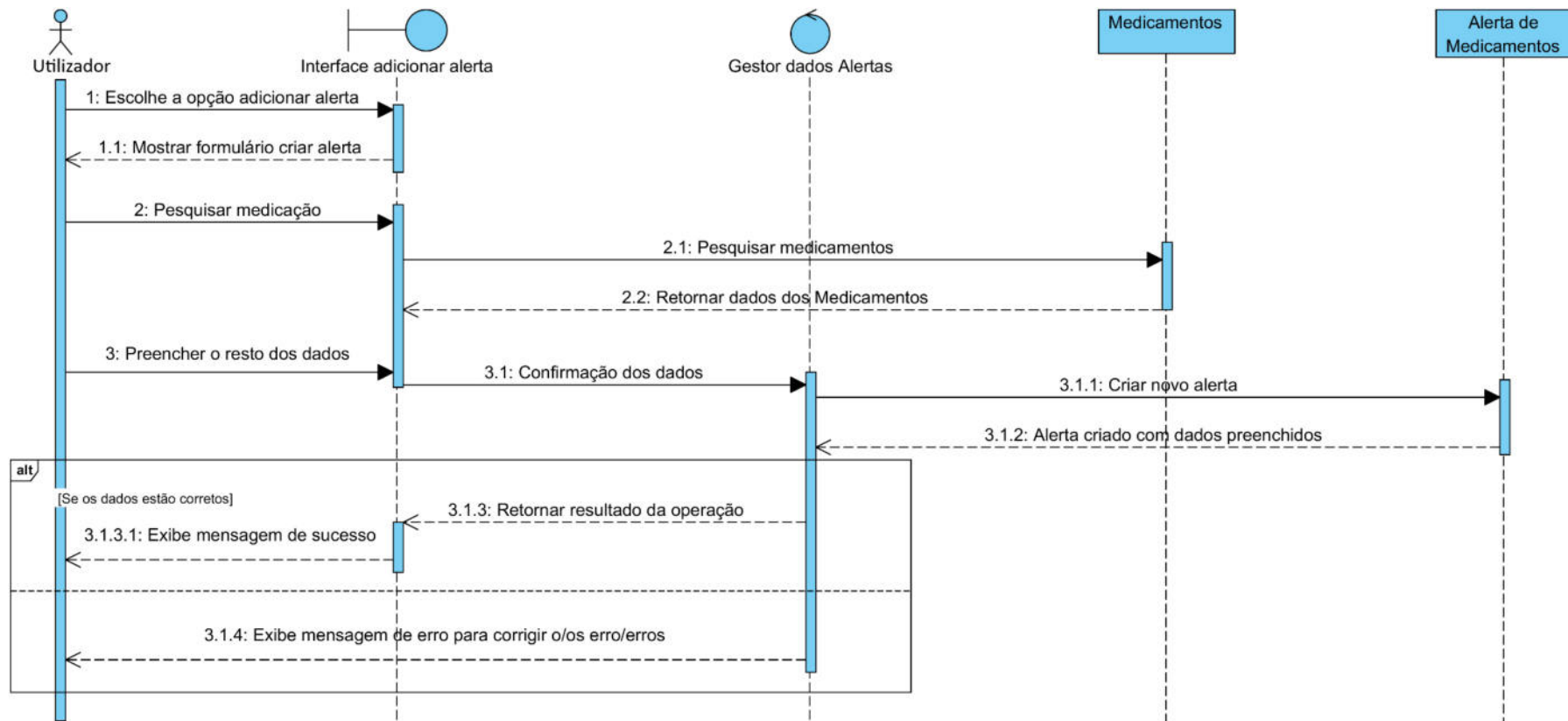


Figura 17 - Sequência de operações – “Adicionar alerta de medicamentos”

3.3.3.3.3. Descrição do caso de uso “Inserir encomenda”

Tabela 7 - Descrição do caso de uso “Inserir encomenda”

Nome	Inserir encomendas
Objetivo	Este caso de uso tem como objetivo inserir encomendas.
Atores	Utilizador
Pré-condição	Farmácias e medicamentos no sistema.
Pós-condição	Registar encomenda no sistema
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a opção “Farmácias”. 2. O ator seleciona a opção “Nova encomenda de medicamentos” 3. O sistema exibe um formulário para preenchimento de uma nova encomenda. 4. O ator preenche o nome da farmácia e do medicamento. 5. O ator clica no botão “Prosseguir encomenda” para confirmar e enviar os dados. 6. O sistema guarda os dados da encomenda. 7. O sistema encaminha o utilizador para o submenu “Farmácias” e retorna mensagem de sucesso de inserção.
Caminho alternativo	<ol style="list-style-type: none"> 4. O ator pretende pesquisar a farmácia e o medicamento <ol style="list-style-type: none"> a) O ator selecionar a opção “Pesquisar farmácia”; b) O ator insere o nome da farmácia; c) O ator selecionar o nome da farmácia; d) O ator selecionar a opção “Pesquisar medicamento” e) O ator insere o nome do medicamento; f) O ator seleciona o nome do medicamento ou os nomes dos medicamentos; g) Voltar para o passo 5 do caminho principal.

Caminhos de exceção	5. a) O ator não preencheu os dados corretamente, retorna para o ponto 4 do caminho principal, sinalizando quais os campos que não foram preenchidos corretamente.
Suplementos ou adornos (requisitos segurança)	Fazer vários testes um de verifique: a) Se o utilizador consegue inserir uma farmácia.

A Figura 18 apresenta o diagrama de sequência do caso de uso “Inserir encomendas” em que a interface “nova encomenda de medicamentos”, o “Gestor dados encomendas” e as tabelas encomendas, medicamentos e farmácias da base de dados estão no do topo do diagrama.

Neste cenário, a interface “nova encomenda de medicamentos” recebe um pedido do “Utilizador” para enviar o formulário de registo de nova encomenda a uma farmácia. A interface responde apresentando o formulário de inserção de encomenda. O “Utilizador” procede à inserção dos dados da encomenda começando pela pesquisa/introdução da farmácia em que pretende fazer a encomenda. A interface “Gestor dados Encomendas” solicita os dados pretendidos à base de dados (tabela farmácias). Esta retorna os dados e o “Gestor de dados Encomendas” filtra-os e apresenta-os na interface “Nova encomenda de medicamentos” e o “Utilizador” seleciona os dados pretendidos. O mesmo procedimento sucede com a pesquisa de medicamentos, mas em vez consultar a base de dados pela tabela farmácia é pela tabela medicamentos. Após o preenchimento de dados necessário para a encomenda os mesmos são submetidos na base de dados.

De um mesmo modo, a interface “Gestor dados encomendas” confirma os dados recebidos e verifica se há existe problema de inserção. Se não for identificado nenhum problema, a interface redireciona os dados para a base de dados (tabela de encomendas). Posteriormente, o “Gestor dados encomendas” trata os dados retornando uma mensagem de sucesso. Caso se regista algum problema, o “Gestor dados encomenda” a aplicação não deixa prosseguir a inserção de dados e emite uma mensagem ao “Utilizador” para corrigir os mesmos.

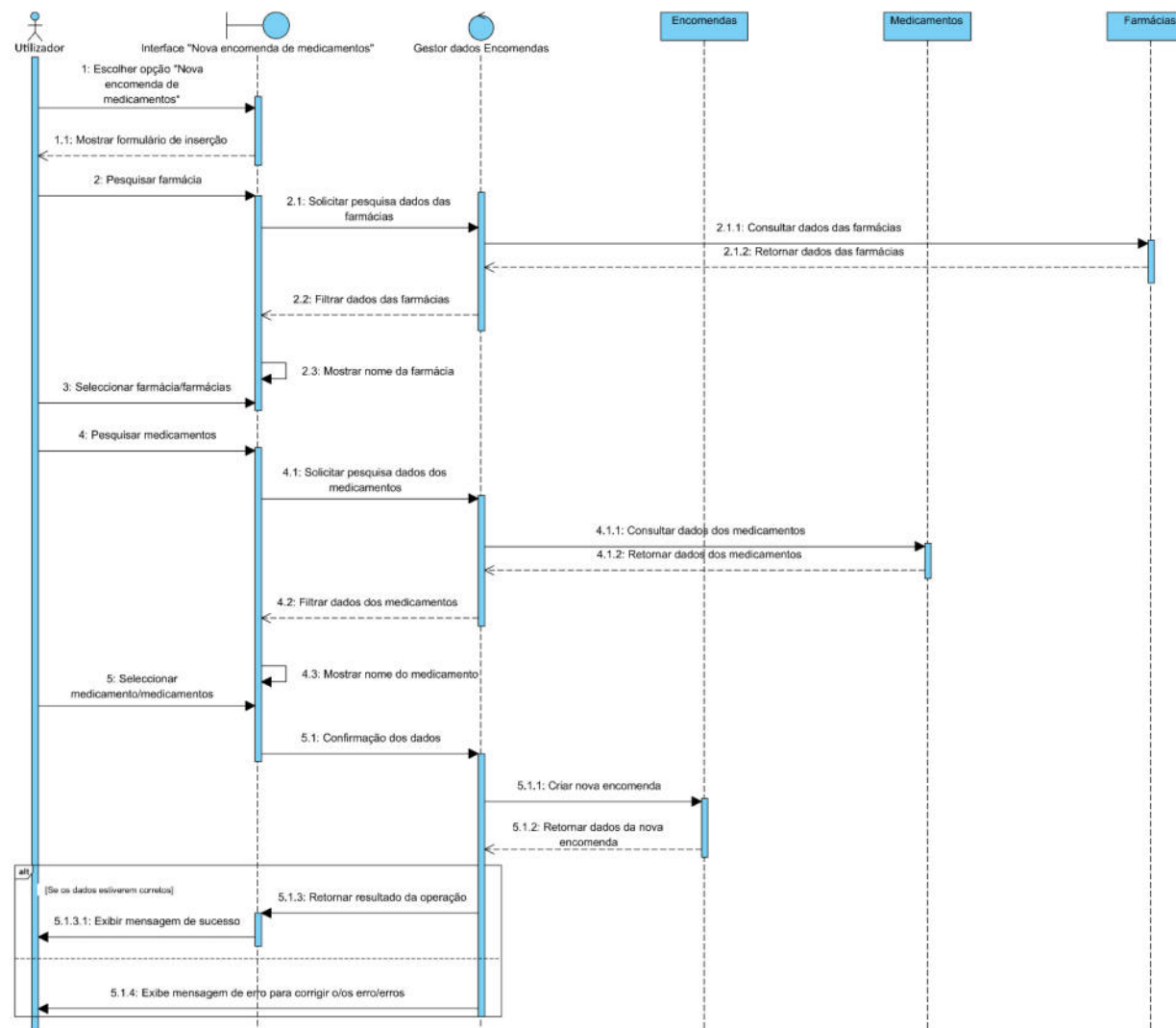


Figura 18 - Sequência de operações – “Inserir encomenda”

3.3.3.3.4. Descrição do caso de uso “Notificação alerta medicamento a utilizar ao utilizador”

Tabela 8 - Descrição do caso de uso “Notificação alerta medicamento a utilizar ao utilizador”

Nome	Notificação alerta medicamento a utilizar ao utilizador
Objetivo	Este caso de uso tem como objetivo alertar o utilizador para tomar a medicação através de notificações.
Atores	Utilizador
Pré-condição	Nenhuma
Pós-condição	Visualização dos alertas no dispositivo.
Caminho Principal	<ol style="list-style-type: none"> 1. O sistema verifica se existe alguma notificação para aquela hora e dia. 2. O sistema lança a notificação. 3. O ator lê e marca como lida.
Suplementos ou adornos (requisitos segurança)	<p>Fazer vários testes onde se verifique:</p> <ol style="list-style-type: none"> a) Que as notificações estão a ser lançadas à hora e dia certo.

A Figura 19 apresenta o diagrama de sequência do caso de uso “Notificação alerta medicamento” em que o dispositivo (“*Local Notifications*”) e as tabelas *Device* e *Alerta Medicamentos* da base de dados se encontram no do topo do diagrama.

Neste cenário a aplicação consulta na base de dados para verificar se o UUID (*Unique Identifier* - Identificador Único Universal) do dispositivo já existe. Caso o mesmo não se encontre, todavia, registado será criado, na base de dados, um novo UUID na tabela *Device*.

Posteriormente, a aplicação vai solicitar outra consulta, que tem como objetivo verificar se existem alertas (tabela alerta medicamentos) para aquele dispositivo. Se existirem retorna os dados e vai informar o dispositivo para configurar as respetivas notificações. Quando for o momento certo o dispositivo vai redirecionar a notificação para a aplicação e acioná-la para o “Utilizador”.

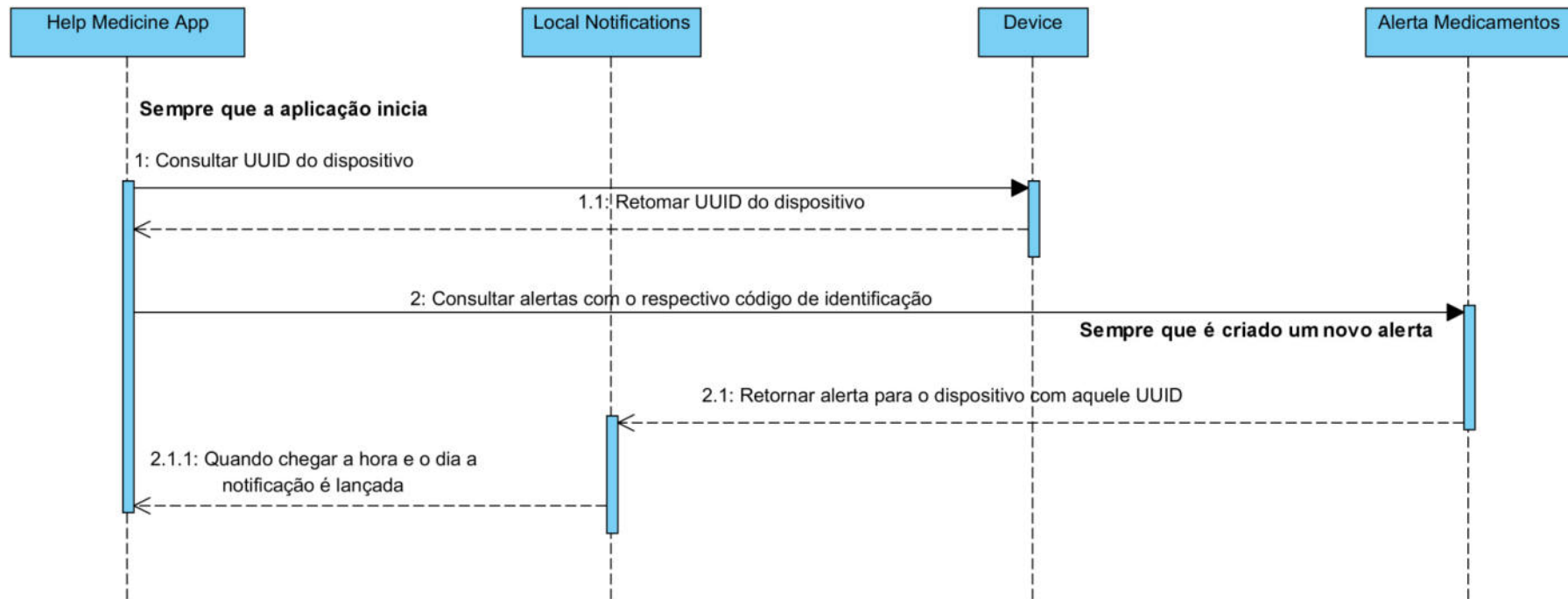


Figura 19 - Sequência de operações – “Notificação de alerta medicamento”

3.3.3.4. Diagrama de Classes

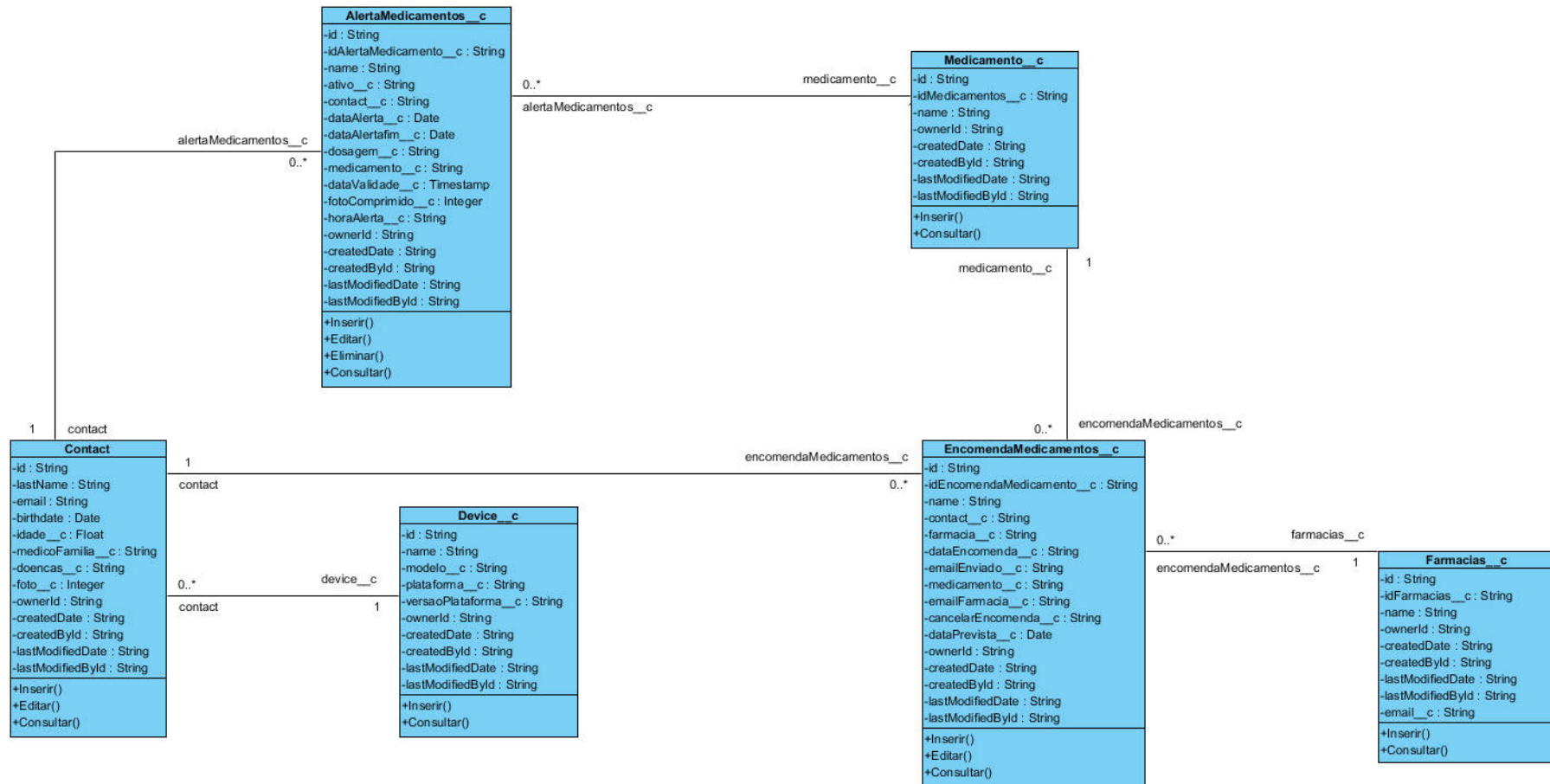
Tendo em conta os casos de uso identificados e os diagramas de sequência, foi definido o Diagrama de Classes da aplicação, apresentado na Figura 20, o qual representa os objetos do sistema e as relações entre os mesmos.

O diagrama de classes tem como finalidade o de fornecer uma visão estrutural do sistema. Para cada objeto é descrita a identidade, os relacionamentos com os outros objetos, os atributos e as operações [34]. A modelação e a criação da base de dados resultam da análise de requisitos, fornecendo dados importantes que serão utilizados posteriormente pelo utilizador [33]. Segue-se uma breve descrição de cada uma das classes identificadas.

A classe “AlertaMedicamentos__c” tem como objetivo registar os dados necessário para alertar o doente para a toma de medicação. É uma classe composta por vários campos do tipo data e hora, atributos essenciais para estabelecer os alertas para a toma de medicação. Também é uma classe que possui um campo representativo do estado do alarme (activo__c) e um campo de inserção de imagem para armazenar a foto do comprimido, se o utilizador assim o entender. Esta classe tem relacionamentos com a classe “Contact__c”, que armazena os dados dos utilizadores e com a classe “Medicamentos”, que armazena os dados relativos aos medicamentos.

Já a classe “EncomendaMedicamentos__c” regista os dados sobre as encomendas em si, isto é para que o utilizador possa fazer uma encomenda de medicamentos a uma farmácia à sua escolha. A classe é composta maioritariamente por campos do tipo data e texto e permitem ao utilizador realizar as operações de inserir, consultar e editar mais facilmente uma encomenda. Relaciona-se com a classe “Farmácias”, que armazena os dados das farmácias e com a classe “Medicamentos__c”.

Por fim, a classe “Device__c”, tem como fim armazenar os dados dos relativos aos dispositivos móveis que instalem a aplicação. Os campos que fazem parte desta classe são os seguintes: id, nome (UUID), modelo, plataforma e versão da plataforma. Relaciona-se com a classe “Contact__c”, que dá a conhecer quais são os utilizadores que existem/utilizam aquele dispositivo móvel.

Figura 20 – Diagrama de classes da aplicação *Help Medicine*

3.3.3.5. Wireframe da aplicação móvel

Tal como referido anteriormente, a análise do projeto passou pela definição de uma *wireframe* ou prototipagem. Podemos definir um *wireframe* como um esqueleto, um protótipo ou uma versão bastante primitiva do visual de um projeto. Este consiste na representação da estrutura do sistema por meio de formas geométricas e linhas. O objetivo principal do wireframe é o auxílio ao gestor de projeto na hora do levantamento de requisitos e também ao programador para alinhar os conteúdos na aplicação [35].

O *wireframe* projetado para o projeto é representado na Figura 21. Foi construído com o *software* o *Balsamiq Mockups*⁵, que produziu uma imagem estática do sistema e possibilitou representar a composição de cada ecrã de maneira rápida e eficiente.



Figura 21 – Wireframe da página inicial do sistema *Help Medicine*

A construção do *wireframe* da aplicação móvel, possibilitou representar a composição de cada ecrã. A utilização desta estratégia permitiu verificar as várias

⁵ <https://balsamiq.com/products/>

ligações entre as funcionalidades da aplicação e perceber o que pode ocorrer quando o utilizador interage com os elementos da interface (Figura 22).

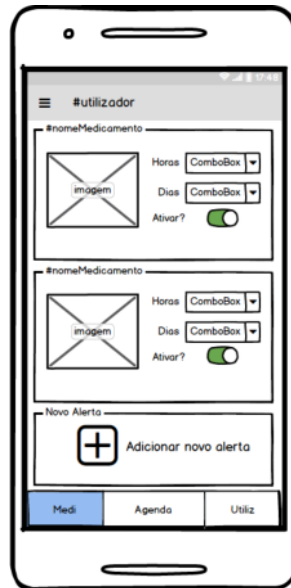


Figura 22 – Wireframe da página alerta de medicamentos do sistema *Help Medicine*

Embora a imagem apresentada não englobe todas as funcionalidades disponíveis na aplicação final, uma vez que ao longo do processo de desenvolvimento foram realizados pequenos ajustes, apresentamos as principais funcionalidades da aplicação. Esta fase de prototipagem serviu de alicerce para o desenvolvimento detalhado do modelo de interface e de interação da aplicação *Help Medicine*.

3.4. Análise de requisitos do sistema embebido

Por cada produto bem-sucedido está uma história de inúmeras tentativas e muito trabalho árduo. A transformação da ideia num produto real envolve mais do que a criação de imagens, sejam elas desenhos manuais ou em computador. O principal objetivo é apresentar o produto de forma rápida, compreensível e tendo em vista os requisitos definidos.

O recurso à prototipagem e algoritmia ajudam a analisar/perceber a funcionalidade dos sistemas em maior profundidade e, por vezes, pode até persuadir e surpreender o cliente. O uso destas técnicas permitem de um modo mais fácil rejeitar um

projeto com base num protótipo e algoritmo, assim como mudar ou aceitar uma nova ideia [36].

Nas secções seguintes, e de modo a definir os requisitos do sistema embebido, é apresentado o protótipo desenvolvido bem como a descrição das etapas envolvidas no seu planeamento.

3.4.1. Protótipo do sistema de alerta de medicamentos

Para o desenvolvimento do protótipo, do sistema embebido, foi necessário desenhar/definir uma estrutura do suporte. Esta teria que ser o mais parecido possível com a realidade, não esquecendo que a mesma teria que ser de fácil utilização e que respondesse aos objetivos propostos, a alerta de toma de medicamentos.

A definição da estrutura do suporte, prototipagem, foi realizada com recurso as *CorelDraw* (Figura 23) e posteriormente guardado para formato *.PNG (Portable Network Graphics)*. Após a realização do protótipo, e após uma pesquisa sobre empresas que realizam trabalho em acrílico procedeu-se ao envio do ficheiro para a empresa que tratou de fazer o suporte em acrílico.

O processo de construção do suporte foi rápido, não obstante surgiram alguns problemas de dimensão que tiveram que ser corrigidas.

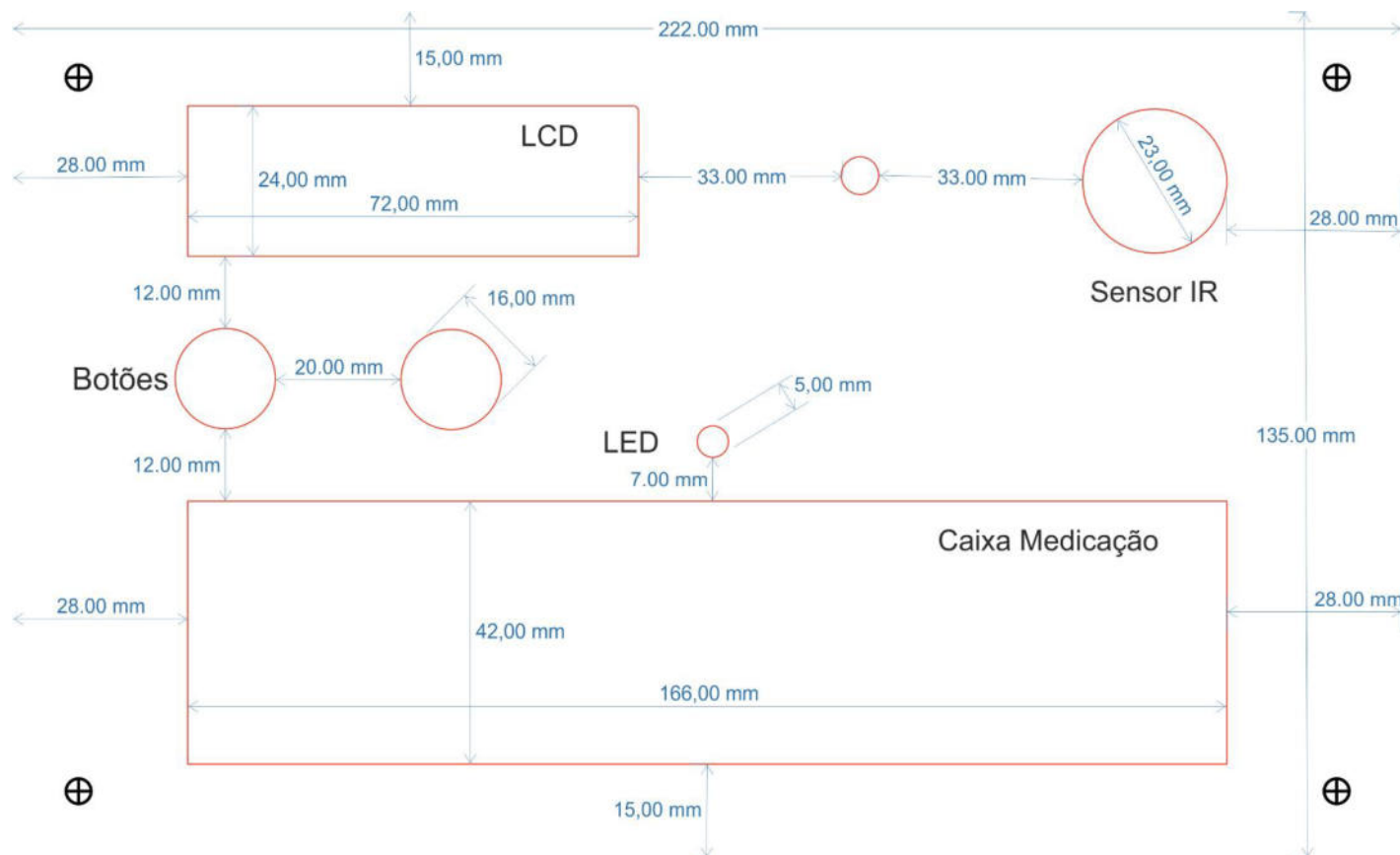


Figura 23 - Desenho do protótipo do sistema de alerta de medicamentos *Help Medicine* em *CorelDraw*

O resultado final, suporte em acrílico, após a construção do desenho apresentado à empresa é exposto na Figura 24.

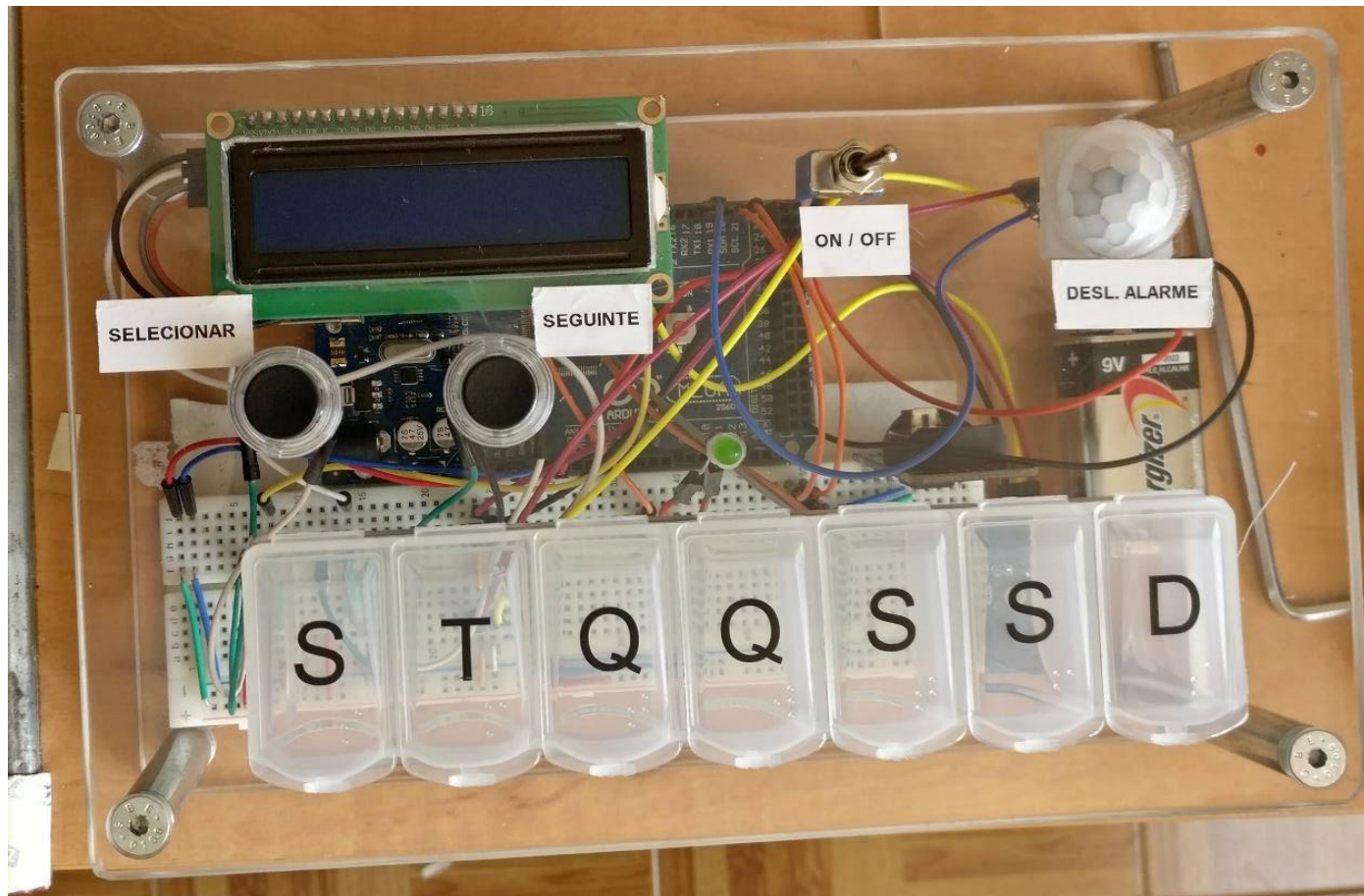


Figura 24 – Protótipo final do sistema de alerta de medicamentos *Help Medicine*

3.4.2. Fluxograma do sistema de alerta de medicamentos

Em simultâneo com a prototipagem, do sistema embebido, alerta de toma de medicamentos, e de modo a que o mesmo respondesse às necessidades iniciais começou-se a pensar em como desenvolver o algoritmo que permitisse o sistema funcionar de forma eficaz e eficiente.

A Figura 25, apresenta o fluxograma estruturado que nos permitiu definir como iríamos executar as tarefas para atingir o objetivo proposto, isto é, que o dispositivo permitisse:

- o utilizador fazer o registo da hora e data da toma da medicação;
- o utilizador deposite os medicamentos nos respetivos compartimentos do dispensador.

O modo como se desenvolveu o algoritmo teve por base, e numa primeira fase, pensar no que o utilizador ao ver o suporte. A opção que nos pareceu mais provável era o utilizador definir as horas, os minutos e o dia da semana do alerta.

No entanto, isto funcionaria apenas se o doente tivesse apenas um medicamento para tomar.

De modo a colmatar, este problema optou-se por perguntar ao utilizador se pretendia criar novo alerta ou então passar para o menu de espera⁶, criando assim duas opções para o utilizador que toma mais ou menos medicação. Não obstante devemos especificar que o sistema foi desenhado para ser possível definir no máximo sete alarmes, pelo menos um por dia por uma semana. Esta opção teve como princípio a não sobrecarga do sistema e a simplicidade de uso por parte do utilizador final.

Quando o utilizador se encontra no menu de espera e se houver mais que um alarme definido, quando um é ativado e posteriormente desligado pelo utilizador o sistema vai continuar no menu de espera, pois existem mais alarmes em espera para serem ativados. Mas, se houver só um alarme definido e já tiver sido ativado ou então já tiverem

⁶ Ecrã que apresenta informações da hora e dia enquanto o utilizador espera pelo alarme para a toma da medicação.

se todos estiverem ativados, o sistema passa automaticamente para o menu de definir alarme para novos alarmes ou então, se o utilizador preferir pode desligar o sistema.

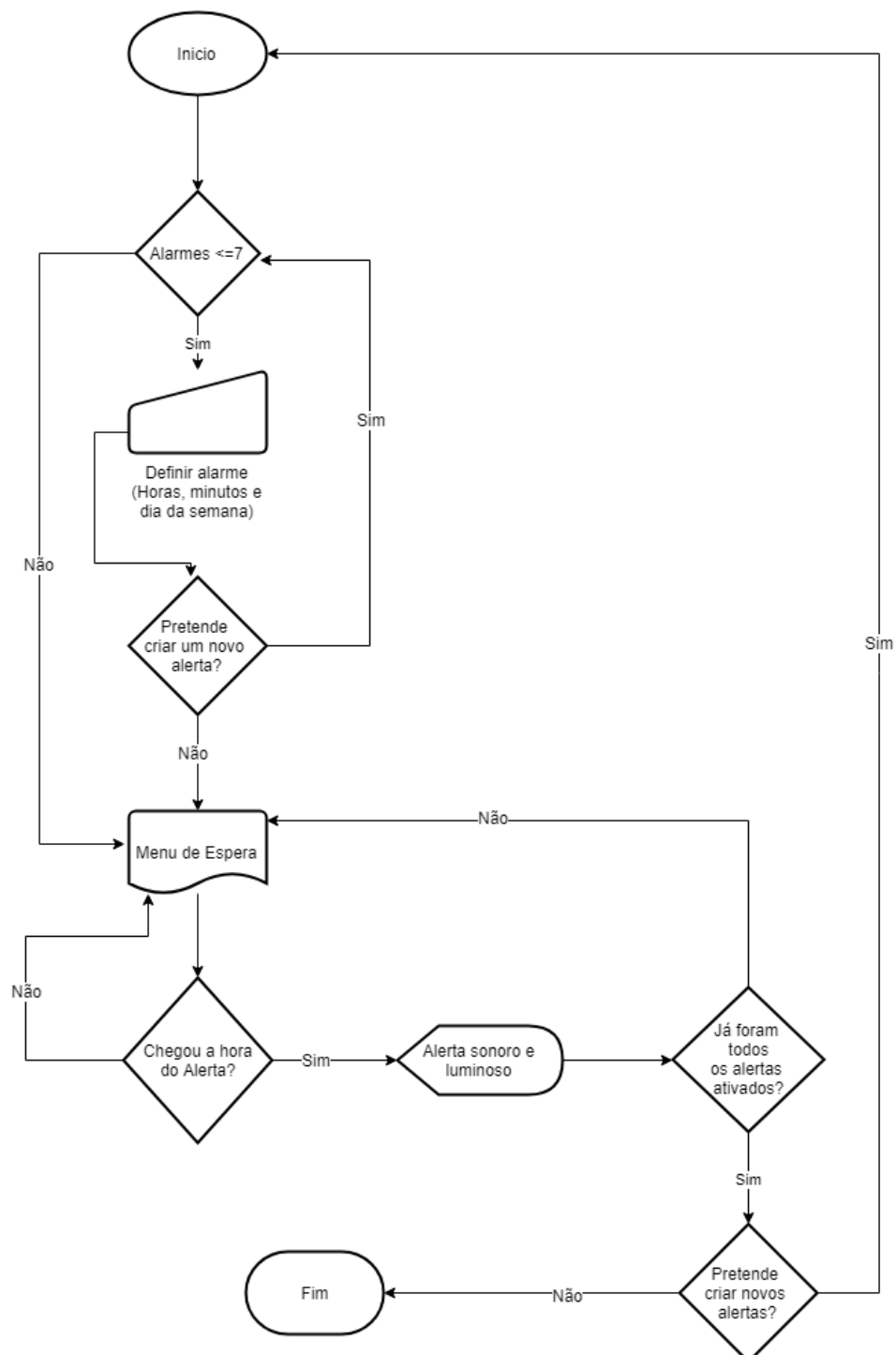


Figura 25 – Fluxograma do sistema de alerta de medicamento *Help Medicine*

4. Desenvolvimento e implementação *Help Medicine*

No presente capítulo pretende-se descrever o processo de desenvolvimento da aplicação móvel *Help Medicine* que tem o propósito de alertar para a toma da medicação o utilizador do smartphone, tendo em vista os requisitos definidos no ponto 3.3. Assim, são abordadas as funcionalidades mais importantes da aplicação, descrevendo para tal as tecnologias utilizadas quer para implementação do projeto bem como para a criação da base de dados, onde são guardados todos os registos inseridos, atualizados, ou eliminados para o correto bom funcionamento do sistema.

Posteriormente descrevemos o processo de desenvolvimento do sistema, etapas e módulos desenvolvidos, de alerta de medicamentos. De um mesmo modo, também serão detalhadas as etapas que permitiram a construção dispositivo de hardware.

4.1. Desenvolvimento da aplicação *Help Medicine*

Tal como anteriormente referido, a aplicação *Help Medicine* tem como principal objetivo permitir que o utilizador tenha a possibilidade de registar a toma de medicação (definir data e hora da toma de cada medicamento) e ser alertado da necessidade de administrar a mesma. A aplicação também permite a visualização, inserção e edição de outros dados tais como medicamentos, farmácias e encomendas. A aplicação permite, também, que vários utilizadores (a pensar no contexto familiar) possam ter acesso ao sistema através de um só dispositivo móvel.

Para o desenvolvimento deste projeto, foi usada a tecnologia cliente/servidor mais conhecida por *webservices*, que permite aceder a serviços. Esta é uma tecnologia flexível e modular conhecida por ser usada em páginas web mais interativas onde existe a comunicação entre vários sistemas/aplicações. Esta arquitetura permite adicionar elementos que aumentam a qualidade e segurança da aplicação a ser desenvolvida, como mecanismos de autenticação, *tokens*, entre outros.

A utilização desta tecnologia permitiu fazer uma separação do *frontend* e do *backend*, fazendo com que seja fácil melhorar a organização do projeto, e ter um melhor controlo em relação à lógica do sistema e ao que é mostrado ao utilizador através do consumo de dados. Além disso, pensou-se que no futuro poderia ser implementado a tecnologia no sistema embebido e por isso não se desenvolveu o *backend* de forma local.

Neste projeto foi usada a linguagem JSON (*JavaScript Object Notation*), tendo esta características como ser de fácil compreensão, completamente independente de qualquer linguagem de programação, e ideal como uma linguagem de troca de informação ou consumo/exposição de dados. Um dos pontos mais importantes pelo uso desta arquitetura é que a lógica feita no *backend*, que pode ser consumida por diversas aplicações, desde web, a mobile, entre outras.

Na Figura 26, está presente um diagrama com a arquitetura de software do sistema onde se mostra como é que o utilizador e os componentes comunicam. O mesmo é composto pela interação dos utilizadores com o *frontend*, sendo este constituído pelo *IONIC Framework* e o *backend*, constituído pelo *Salesforce*

O processo de comunicação passa pelo utilizador fazendo pedidos no *frontend*. Estes pedidos são realizados através dos controladores, que por sua vez vão comunicar com o *provider Angular Http Client* que permite a aplicação aceder e manipular partes da tecnologia HTTP (*Hypertext Transfer Protocol*) como os *requests* e os *responses*. Este também efetua o pedido ao *Salesforce (Apex Rest)* indicando qual é a *route* que pretende aceder para obter dados.

O *Apex Rest* recebe o pedido e verifica a *route* que lhe é enviada. Após verificar a mesma, realiza a lógica que essa *route* possui tais como verificação da *api-key* e *api-client*, e por fim o pedido é enviado à base de dados do salesforce através de um *query*.

Por fim, a base de dados recebe a *query*, executa-a e envia o resultado para o *Apex Rest* que por sua vez devolve os dados ao *provider* do *IONIC Framework* que identifica onde estes serão apresentados os dados ao utilizador.

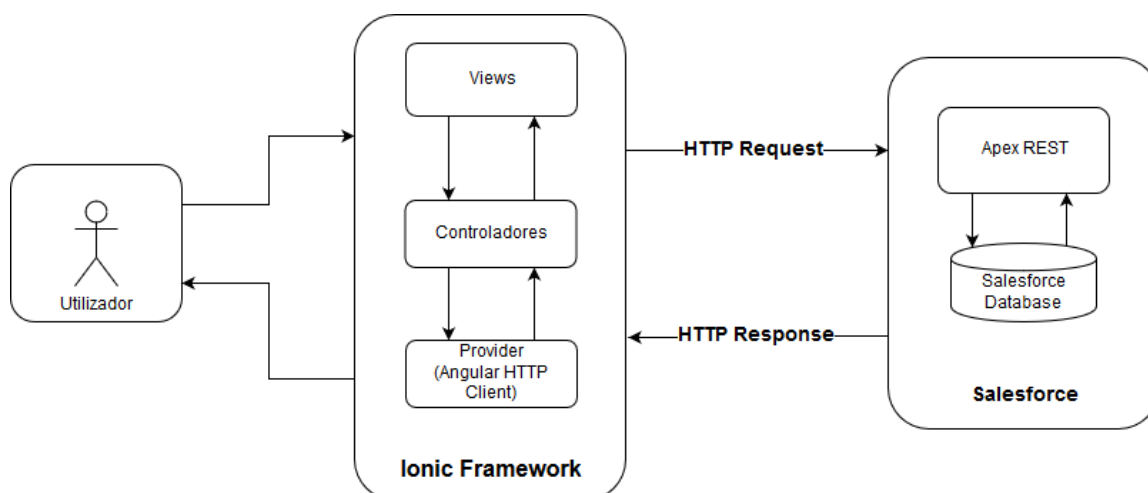


Figura 26 – Arquitetura do sistema da aplicação móvel *Help Medicine*

De seguida são apresentados os pontos principais do desenvolvimento do *backend* e do *frontend* que desempenham um papel fundamental ao bom funcionamento da aplicação.

4.1.1. Desenvolvimento do *webservice*

Para o desenvolvimento do *backend*, foi utilizada a linguagem de programação APEX que permite executar instruções no *Salesforce*.

A APEX da Salesforce é uma linguagem de programação orientada a objetos (POO), com sintaxe semelhante a Java e desenvolvida a pensar na integração com as restantes tecnologias da plataforma *Force.com*. Desta forma, foi possível utilizar instruções de *Data Manipulation Language* (DML), processar registos em massa com controlo transaccional e criar *triggers* sem ser necessária a inclusão de bibliotecas e a configuração de ligações à base de dados. Para além disso, contém instruções normalmente encontradas neste género de linguagens, como controlo condicional (*if-else*), looping (*do-while*, *while* e *for*), invocação de métodos e tratamento de exceções.

Além disso, foram utilizadas funcionalidades disponíveis pela plataforma, como por exemplo, a segurança do *backend* e o envio de *emails* e acrescentou-se os métodos necessários (GET, PATCH, POST, DELETE) para a aplicação consumir os dados pretendidos.

4.1.1.1. Métodos para requisição de dados

Método GET

Caso seja um pedido de visualização de dados é ativada a operação GET. Este método requisita os dados pretendidos à base de dados. A resposta a este pedido é obtida em formato JSON, que contém todas as informações que o utilizador requisitou. Apresenta-se a seguir um exemplo de uma das várias respostas JSON utilizadas na aplicação móvel para pedidos de dados, neste caso, os alertas de medicamentos existentes.

```
@HttpGet
global static void getData() {
    RestRequest request = RestContext.request;
    RestResponse res = RestContext.response;
    res.addHeader('Content-Type', 'application/json');

    String xclient = request.headers.get('api-client');
    String xkey = request.headers.get('api-key');

    if(verificar(xclient, xkey)){
        try{
            List<AlertaMedicamentos__c> alerta = [SELECT Name,
dataAlerta__c, dataAlertafim__c, horaAlerta__c, dosagem__c,
Ativo__c, Contact__c, Medicamento__c from AlertaMedicamentos__c];
            res.responseBody = Blob.valueOf(JSON.serialize(alerta));
            res.statusCode = 200;
        } catch (Exception e){
            res.responseBody = Blob.valueOf(e.getMessage());
            res.statusCode = 500;
        }
    } else {
        res.statusCode = 401;
        res.responseBody = Blob.valueOf(JSON.serialize('API KEY
errada!'));
    }
}
```

Figura 27 – Pedido para ver todos os alertas de medicamentos existentes.

Método POST e PATCH

Para adicionar dados ao *Webservice*, é utilizado o método POST que permite enviar dados anexados no corpo do URL para armazenamento na base dos dados. Na Figura 28, é apresentado um exemplo do código desenvolvido para inserção de um alerta para um medicamento.

```
@HttpPost
```

```

global static void create () {
    RestRequest request = RestContext.request;
    RestResponse res = RestContext.response;
    res.addHeader('Content-Type', 'application/json');

    String xclient = request.headers.get('api-client');
    String xkey = request.headers.get('api-key');

    if(verificar(xclient, xkey)){
        try{
            Map<String, Object> fieldMap = (Map<String,
Object>)JSON.deserializeUntyped(request.requestBody.toString().trim()
);
            Schema.SObjectType targetType =
Schema.getGlobalDescribe().get('AlertaMedicamentos__c');
            SObject newSobject = targetType.newSObject();
            Map<String, Schema.SObjectField> targetFields =
targetType.getDescribe().fields.getMap();
            for (String key : fieldMap.keySet())
            {
                Object value = fieldMap.get(key);
                Schema.DisplayType valueType =
targetFields.get(key).getDescribe().getType();
                if (value instanceof String && valueType !=
Schema.DisplayType.String)
                {
                    String svalue = (String)value;
                    if (valueType == Schema.DisplayType.Date)
                        newSobject.put(key,
Date.valueOf(svalue));
                    else if(valueType == Schema.DisplayType.Time)
                        newSobject.put(key, ParseTime(svalue));
                    else if(valueType ==
Schema.DisplayType.DateTime)
                        newSobject.put(key,
DateTime.valueOfGmt(svalue));
                    else if (valueType ==
Schema.DisplayType.Percent || valueType ==
Schema.DisplayType.Currency)
                        newSobject.put(key, svalue == '' ? null :
Decimal.valueOf(svalue));
                    else if (valueType ==
Schema.DisplayType.Double)
                        newSobject.put(key, svalue == '' ? null :
Double.valueOf(svalue));
                    else if (valueType ==
Schema.DisplayType.Integer)
                        newSobject.put(key,
Integer.valueOf(svalue));
                    else if (valueType ==
Schema.DisplayType.Base64)
                        newSobject.put(key,
Blob.valueOf(svalue));
                    else
                        newSobject.put(key, svalue);
                }
            }
            else
                newSobject.put(key, value);
        }
    }
}

```

```
        }
        insert newSobject;
        res.statusCode = 200;
        res.responseBody =
Blob.valueOf(JSON.serialize(newSobject));
    } catch (Exception e) {
        res.responseBody =
Blob.valueOf(json.serialize(e.getMessage()));
        res.statusCode = 500;
    }
} else {
    res.statusCode = 401;
    res.responseBody = Blob.valueOf(JSON.serialize('API KEY
errada!'));
}
}
```

Figura 28 – Inserção de um novo alerta de medicamento.

Após a receção dos dados no *webservice*, este identifica os vários campos e valores e insere-os na base de dados. Posteriormente é enviado a resposta em formato JSON para a aplicação com o estado da operação (sucesso ou insucesso).

O mesmo processo é realizado com a atualização dos dados (PATCH). A única diferença é que além de anexar os dados a alterar no corpo do URL também é enviado como parâmetro um valor identificativo do campo que pretendemos alterar. Na Figura 29, é apresentado o URL de alteração de dados do utilizador. Neste caso, passa-se como parâmetro o valor do id do utilizador que se pretende alterar.

<https://--/services/apexrest/Utilizadores?id=idDoUtilizador>

Figura 29 – Exemplo de um pedido de alteração de dados

Em relação à receção de dados, este tem o mesmo comportamento que já teve anteriormente com o método POST com a exceção que vem um parâmetro novo e por isso necessita de identificar qual é o valor que o utilizador pretende alterar. Após a alteração realizada, este envia uma resposta JSON com o estado da operação (sucesso ou insucesso).

Método DELETE

Quando é necessário apagar dados, é utilizado o método DELETE. Identifica-se qual o campo que se pretende apagar e é enviado no URL como parâmetro. Na Figura

30, é apresentado o URL para apagar alertas. É enviado como parâmetro o id do alerta que pretendemos apagar.

<https://--/services/apexrest/Alertas?id=idDoAlerta>

Figura 30 – Exemplo de um pedido de apagar dados

O *webservice* recebe o pedido que o utilizador pretende eliminar dados, identifica qual o id e elimina os valores que estão relacionados com aquele id. Se tudo correr bem, este envia uma resposta com a operação sucesso para a aplicação móvel.

4.1.1.2. Envio de *emails*

Tendo em conta que a aplicação foi pensada para ter uma componente de encomenda de medicamentos a farmácias, houve a necessidade de criar um processo que envia um *email* às farmácias para requisição de medicação pretendida pelo paciente.

O processo de envio de *email* para a farmácia, consiste em fazer um pedido ao *backend* que verifica se é a *route* de encomenda de medicação. Após a verificação da *route* e respetiva inserção da encomenda na base de dados, *este* envia o *email* à farmácia que o utilizador pretende.

A Figura 31 apresenta um *email* enviado pelo paciente à farmácia encomendando alguns medicamentos.



Figura 31 – Exemplo de um *email* enviado pelo *webservice* à farmácia

Para fazer a deteção dos dados a serem enviados no *email* nível do *backend*, são utilizadas as funcionalidades que a plataforma do *Salesforce* disponibiliza tais como:

Email Alert e Process Builder. No *Email Alert*, foi configurado, Figura 32, o *template* que seria utilizado para envio do *email*. A Figura 31 exemplifica qual é a formatação e o corpo do *email* que foram criados bem como o *email* da farmácia. No *Process Builder*, foi realizado um algoritmo, que permite verificar quando é necessário proceder ao envio do *email*, Figura 33.

Figura 32 – Configuração do *EmailAlert* no *Salesforce*

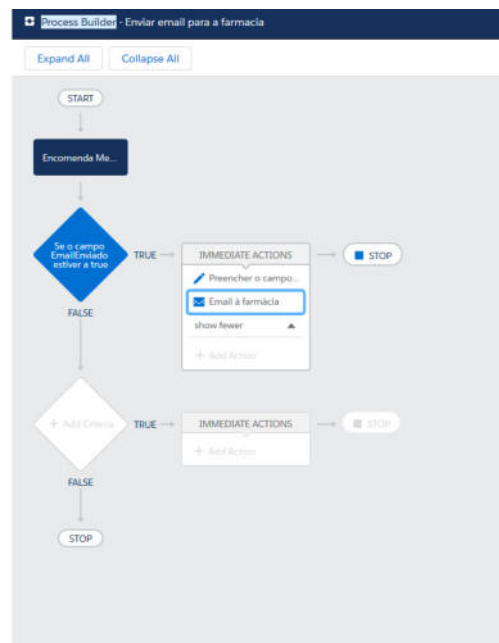


Figura 33 – Desenvolvimento do algoritmo para envio de *emails*

O mesmo processo foi realizado para confirmação do *email* do utilizador. Este processo não tinha sido pensado inicialmente, mas após o desenvolvimento do *backend* verificou-se esta necessidade porque sendo o *email* um dado sensível e pessoal, é importante existir uma confirmação de que o mesmo foi enviado com sucesso. O mesmo sucede quando se regista um novo alerta de medicação. O paciente recebe no seu *email* uma confirmação que criou um novo alerta para determinado dia e hora.

4.1.2. Desenvolvimento do *frontend*

Para desenvolvimento do *frontend* foi utilizada a *framework* *Ionic*. Esta, permitiu que o desenvolvimento fosse multiplataforma, ou seja, estaria a desenvolver para *Android* e ao mesmo tempo para *IOS* e por isso houve um menor tempo de “construção da aplicação”. Isto porque há apenas uma base de código que será utilizada para gerar os apps para ambas as plataformas.

Além disso, o desenvolvimento dos ecrãs é um processo relativamente fácil. A *framework* apresenta um conjunto de *tags* que se ajustam a cada plataforma, poupando o trabalho de desenvolver aquele componente em específico para *Android* e *IOS*.

Por fim, a documentação da *framework* é bem estruturada e apresenta uma grande comunidade e bastante ativa.

Apresentam-se de seguida as fases onde vão ser abordadas algumas das interfaces mais importantes do projeto. O objetivo é o apresentar as várias funcionalidades presentes em cada uma delas bem como a interação que os utilizadores irão ter com as mesmas.

4.1.2.1. Ecrã inicial da aplicação

A Figura 34, representa o ecrã inicial da aplicação, que é apresentada ao utilizador assim que este acede à aplicação. O objetivo principal desta interface é dar as boas vindas ao utilizador da aplicação. Este ecrã inicial permite ainda a possibilidade de criar um novo utilizador ou aceder a um utilizador já existente podendo posteriormente aceder ao ecrã de alertas, ecrã de medicamentos, entre outros.

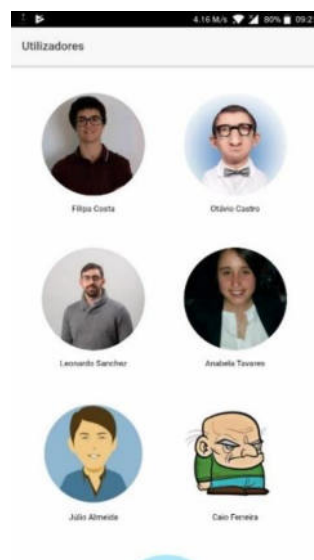


Figura 34 – Ecrã inicial da aplicação *Help Medicine*

4.1.2.2. Ecrã alerta de medicamentos

No que diz respeito à alerta, de toma de medicamentos, Figura 35, o sistema apresenta ao utilizador vários tipos de alertas disponíveis, bem como toda a informação relativa a essa toma de medicamento (nome do medicamento, datas de início e fim, qual a dosagem que deve tomar, e se o alerta está ativo).

Ao registar um novo alerta, Figura 36, o sistema permite, de um modo ágil, e com o objetivo de tornar o processo mais fácil pesquisar o medicamento, que deverá ficar no alerta, através da opção *Auto complete* através da caixa de texto “Nome medicamento” (Figura 37).

Quando o utilizador tiver finalizado a inserção de dados relativo a uma Alerta no sistema o mesmo, é inserido no *backend* e posteriormente colocado em *background* na aplicação. Esta ação permitirá que o alerta fique disponível ao utilizador, através de uma notificação, na hora e dia pretendido, Figura 38.



Figura 35 – Ecrã dos alertas criados para a toma de medicação



Figura 36 – Ecrã de registo do alerta de toma de medicação



Figura 37 – Auto-complete do ecrã de registo do alerta

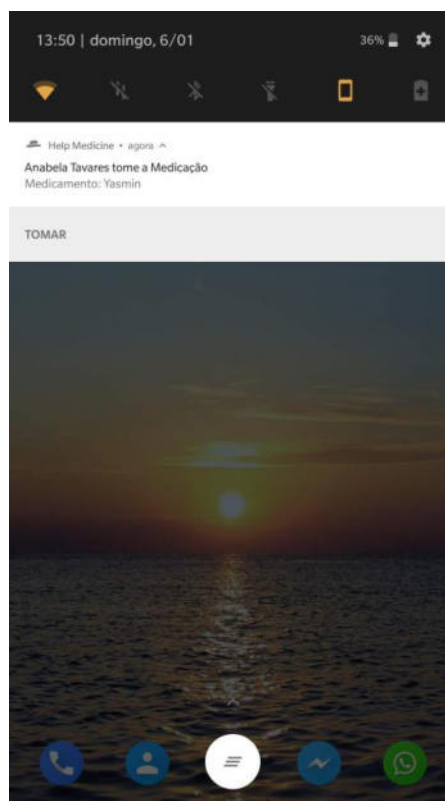


Figura 38 – Notificação de alerta ao paciente para tomar a medicação.

4.1.2.1. Ecrãs de encomenda de medicamentos

Tal como especificado, na definição dos requisitos o sistema deverá ser capaz de enviar encomenda de medicamentos, para uma farmácia escolhida pelo utilizador.

Para efetuar um encomenda, o utilizador deve selecionar a opção encomendas, na sequência de ação é-lhe apresentado o ecrã detalhes das encomendas dos medicamentos, Figura 39, onde são apresentados os detalhes das encomendas, tais como a medicação encomendada, a farmácia e a data de encomenda. O utilizador poderá se assim o desejar cancelar a encomenda ou então adicionar uma nova encomenda. Caso pretenda cancelar a encomenda, será enviado um *email* para a farmácia a cancelar a mesma. Não obstante, a aplicação permite ao utilizador voltar a ativar a encomenda cancelada reativando-a.

Para adicionar uma nova encomenda o utilizador poderá efetuara a mesma através da pesquisa de farmácia e medicamento, Figura 40. Após a inserção e validação dos dados a aplicação envia um *email* para a farmácia com os dados da encomenda pretendida (processo descrito na secção 4.1.1.2, Envio de *emails*).



Figura 39 – Ecrã detalhes das encomendas do paciente

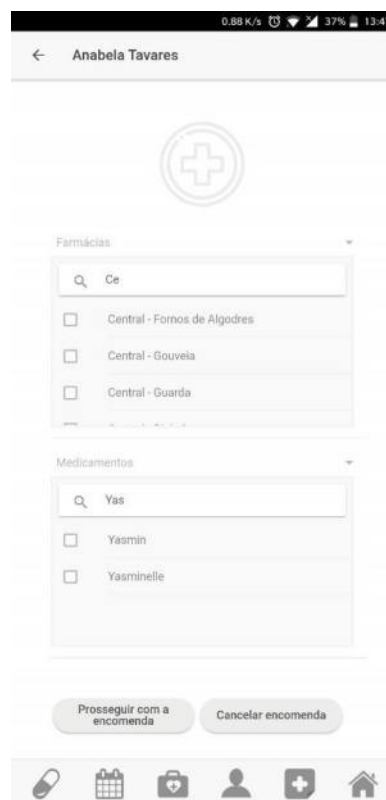


Figura 40 – Ecrã encomenda de medicamentos.

4.2. Desenvolvimento do sistema embebido de alerta de medicamentos

O segundo sistema de alerta de medicamentos desenvolvido, *Help Medicine*, foi construído com base na plataforma *open-source* Arduíno Mega, recorrendo a um conjunto de módulos de hardware específicos (*protoboard*, *LEDs*, *push button*, *LCD*, fios de ligação, *buzzer*, sensor IR infravermelho, *switch*) e um dispensador de medicação. A construção deste segundo sistema teve por base a possibilidade de que outros utilizadores, não detentores de dispositivos móveis ou tendo um dispositivo móvel apresentem

difficuldade na sua utilização, como é o caso de algumas pessoas idosas ou com alguma doença específica. De um modo mais simplista o sistema permite que o seu utilizador possa de forma fácil configurar alertas para a toma de medicamentos, definindo a data, hora e dia da semana da respetiva toma de cada medicamento.

4.2.1. Arquitetura do sistema embebido *Help Medicine*

Com base nos módulos de hardware, anteriormente apresentado, foi desenhado um diagrama geral do sistema embebido *Help Medicine* que é apresentado na Figura 41 – Arquitetura do sistema embebido Esta figura representa, de uma forma simplificada, todos os módulos e componentes do sistema e a forma como estes interagem entre si.

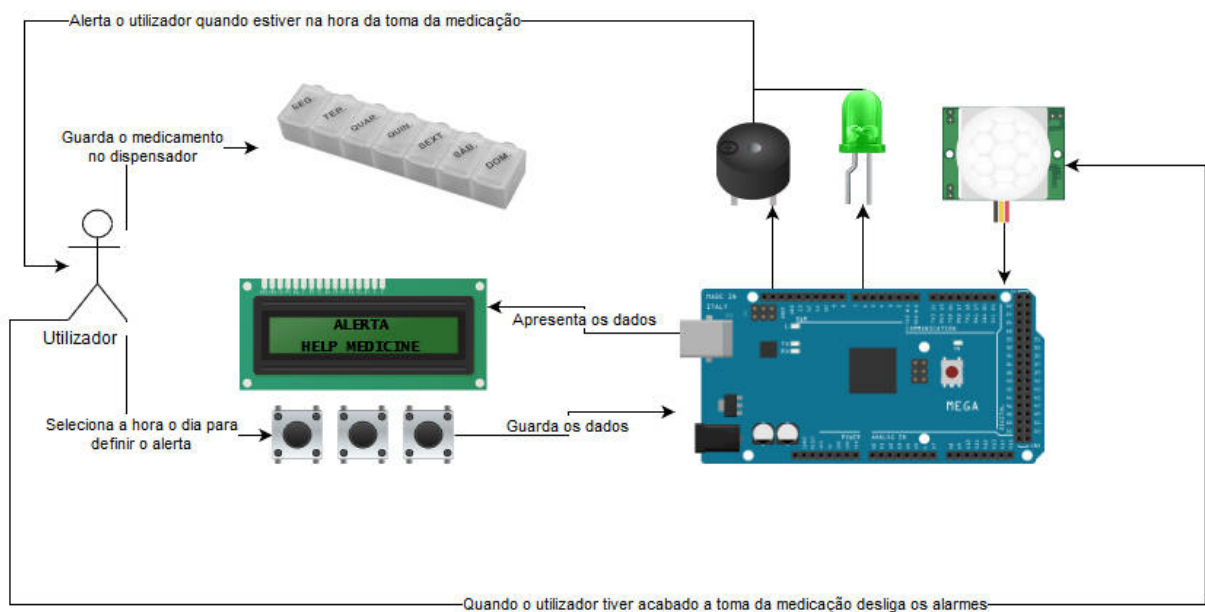


Figura 41 – Arquitetura do sistema embebido

O dispositivo está pronto a ser utilizado quando o utilizador faz o respetivo registo da hora e data da toma da medicação e deposita os medicamentos nos respetivos compartimentos do dispensador.

4.2.2. Módulos do dispositivo *Help Medicine*

No que respeita à vertente eletrónica do hardware do sistema, o dispositivo, ilustrado na Figura 42, é constituído pelos seguintes blocos: alimentação (*switch*), microcontrolador, alarmes luminosos (LEDs), alarmes sonoros (*buzzer*), display com informações úteis ao utilizador (LCD), botões para definir os alertas, parar alarme sonoro (sensor infravermelho) e um dispensador de medicamentos.

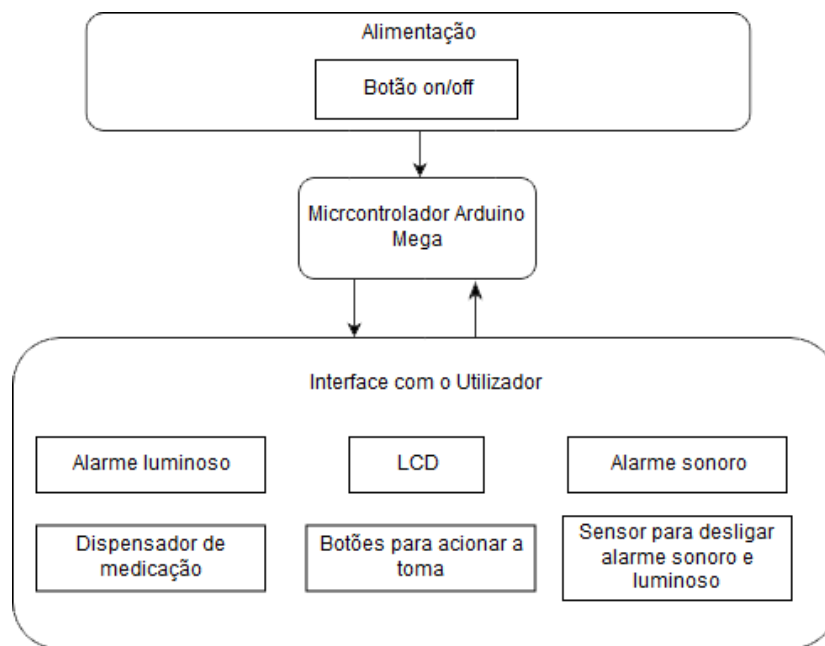


Figura 42 – Diagrama de blocos do sistema embebido

De seguida far-se-á uma descrição detalhada dos módulos que compõem o sistema utilizados no projeto. A nível da interface com o utilizador, serão abordados, respetivamente, o *push button*, o LED, o LCD, buzzer e o sensor.

4.2.2.1. Push Button

O *push button* (ou botão de pressão) é um componente que conecta dois pontos de um circuito quando é pressionado. Tem o mesmo funcionamento elétrico que um interruptor, fechando ou abrindo o circuito elétrico. Neste projeto, o *push button* é

definido como uma entrada e é pressionado pelo utilizador para que este selecione o horário e dia de toma da medicação. O *push button* utilizado neste projeto tem o aspeto representado na Figura 43 [37].



Figura 43 – Representação *push button* utilizado no projeto [38]

4.2.2.2. LED

O LED (*Light Emitting Diode*) é um componente eletrónico semicondutor, ou seja, um díodo emissor de luz que tem a propriedade de transformar energia elétrica em luz visível. O LED é um componente do tipo bipolar, ou seja, tem um terminal chamado ânodo e outro chamado cátodo. Dependendo de como for polarizado, permite ou não a passagem de corrente elétrica e, conseqüentemente, a geração ou não de luz. No sistema Arduino o LED é definido como uma saída e, neste projeto, o LED é acionado quando chega a hora exata da toma da medicação. A Figura 44, ilustra uma representação simbólica e esquemática de um LED [39].

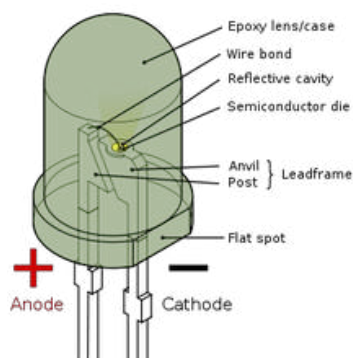


Figura 44 – Representação esquemática de um LED [40]

4.2.2.3. LCD

Um LCD (*Liquid Crystal Display*) é um painel fino usado para exibir informações por via eletrônica, como texto ou pequenas imagens. Sempre que um LCD é utilizado com a plataforma Arduino recorre-se, neste caso, à biblioteca *LiquidCrystal_I2C.h*. De salientar também que um display LCD necessita de estar conectado a 4 pinos do Arduino, enquanto LEDs ou *push buttons* necessitam de apenas uma porta. Neste projeto o LCD é permite que o utilizador consiga visualizar a hora e o dia em que o utilizador está a registar o alerta ou a tomar o medicamento [41].



Figura 45 - Representação de um LCD adaptável para utilização no Arduino [42]

A Figura 45, ilustra uma representação de um LCD adaptável para utilização na plataforma de desenvolvimento Arduino. Existem vários formatos de LCD, mas este foi o LCD escolhido para utilização neste projeto, uma vez que consideramos que o mesmo possibilita uma correta representação da informação.

4.2.2.4. Buzzer

Um *Buzzer* nada mais é do que um pequeno altifalante capaz de emitir sons em diversas frequências. O *buzzer* é normalmente usado em projetos que necessitam de avisos sonoros, relógios com alarme e/ou até para reproduzir músicas. Neste projeto o *buzzer* é acionado quando chega a hora exata e predefinida pelo utilizador, da toma da medicação. A Figura 46, ilustra o *buzzer* utilizado no sistema de alerta de medicação.

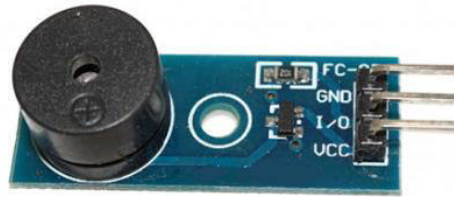


Figura 46 – Representação do Buzzer utilizado no projeto [43]

4.2.2.5. Sensor Infravermelho

O sensor infravermelho (IR) possui um circuito transmissor e um recetor posicionado um ao lado do outro. Quanto um objeto ou uma pessoa passa em frente ao sensor, o sinal IR é refletido e detetado pelo recetor, que coloca o pino de saída em nível baixo (LOW) e aciona um led localizado na parte traseira do sensor. Neste projeto, o sensor IR, é utilizado para deteção de movimentos no ambiente. Este sensor permite identificar com maior precisão quando é que o utilizador pretendia parar o sinal sonoro e luminoso do alerta de forma a que o utilizador nem percebesse que estava a dar “ordens” a um computador (ubíqua). A Figura 47, mostra o sensor utilizado no projeto.



Figura 47 – Representação de um Sensor infravermelho utilizado no projeto

4.2.2.6. Dispensador de medicação

Aquando da definição de como seria definido o dispensador de medicamentos optou-se por utilizar um tabuleiro de tomas de formato retangular, compartimentado com

7 secções. Este tabuleiro de tomas deve ser preenchido, pelo utilizador, sempre que este desejar ser avisado de tomar a medicação. A Figura 48, ilustra a caixa de medicamentos que foi projetada e definida para o projeto.



Figura 48 – Caixa de Medicamentos semanal, utilizada no projeto

5. Testes e resultados

Neste capítulo, são apresentados todos os resultados obtidos após o desenvolvimento do sistema *Help Medicine*. O capítulo apresenta, também, alguns testes de validação que foram realizados com o objetivo de assegurar que o software e o hardware desenvolvidos tenham o funcionamento e o desempenho esperado e acordo com os objetivos inicialmente definidos.

5.1. Validação do sistema *Help Medicine*

Com o objetivo de verificar o bom funcionamento do dispositivo *Help Medicine* foram efetuados um conjunto de testes de validação. Estes testes consistiram em o dispositivo permanecer na casa de uma pessoa durante algum tempo e este registrar as falhas e o sucesso que o dispositivo revelou.

A finalidade destes testes, para além de validar o protótipo desenvolvido, também consiste em garantir que os sistemas estão funcionais para integrar um ambiente assistido de uma pessoa.

5.1.1. Características da amostra

Como se trata da validação de um conceito e não de estudos estatísticos, o elevado tamanho da amostra não se mostra muito importante para a validação deste dispositivo. Tendo isso em conta, para a utilização do dispositivo *Help Medicine* e consequente realização dos testes de validação foram convidadas indivíduos com idades compreendidas entre os 23 anos e os 62 anos. Existem 3 tipos de indivíduos na amostra.

- os que tomam medicação diária, desde 2 a 4 medicamentos
- os que revelam alguns problemas de esquecimento

- os que não necessitam de medicação diária, mas estavam numa fase em que tinham de tomar medicação e precisam de um alerta para a toma da mesma.

Todas as características relevantes da amostra podem ser consultadas na Tabela 9.

Tabela 9 - Detalhes da amostra da população

Individuo	Género	Idade	Prob. Memoria	Medicação	Escolaridade
1	M	33	Não	1 comp	12º Ano
2	M	27	Não	1 comp	Licenciatura
3	M	27	Não	1 comp	12º Ano
4	M	62	Sim	2 comp	6º Ano
5	F	60	Não	4 comp	4º Ano

Antes da utilização do *Help Medicine* foi explicado de forma detalhada, a cada um dos indivíduos da amostra, o funcionamento do dispositivo, tanto a nível de hardware como de software.

Nesta fase inicial de apresentação do dispositivo, alguns dos indivíduos mostraram alguma relutância devido, muito provavelmente, ao desconforto com o uso de tecnologia a que não estão acostumados, esse problema foi mitigado quando lhes foi apresentado um cuidador, que os iria acompanhar e ajudar caso ocorresse algum problema.

A autora do presente projeto ficou responsável por ser a cuidadora dos indivíduos mais relutantes, encarregando-se das seguintes tarefas: definir os alarmes e preencher o tabuleiro de tomas do *Help Medicine* com os respetivos medicamentos. Para além disso, na aplicação móvel, esta ficou também de preencher os dados, receber no seu e-mail todas as notificações de cada um deles e alertar o doente, acerca do que fazer aquando a altura da toma da medicação.

5.1.2. Resultados e Discussão

Após cada um dos indivíduos usufruir dos sistemas durante um período de tempo foi-lhes pedido um *feedback* através de um formulário, presente no Anexo C – Formulário de validação do *Help Medicine*, para que assim se pudessem tirar conclusões acerca da usabilidade do dispositivo.

Após a análise das conclusões, verificou-se que os indivíduos da amostra consideraram que o *Help Medicine* os ajudou num estilo de vida mais autónomo. Todos os indivíduos revelaram que o dispositivo nunca mostrou falhas e alertou-o sempre a tempo da toma da medicação. Com estes dados podemos concluir que os indivíduos se adaptaram bem ao uso dos sistemas apresentados, mesmo aqueles que inicialmente se mostraram mais reticentes na hora da sua utilização.

De uma maneira geral, conclui-se que dentro de uma amostra de 5 indivíduos todos consideraram que o uso dos sistemas *Help Medicine*, os pode ajudar na gestão da sua medicação e num estilo de vida menos dependente de terceiros.

Acerca da validação do protótipo *Help Medicine* como funcional, era importante que a quantidade de falhas deste dispositivo fossem as mais reduzidas possíveis. Existem algumas situações que são consideradas falhas, nomeadamente quando o protótipo não deixa apagar ou editar alarmes, assim como definir mais que 7 alarmes ou quando os fios de ligação, por erro humano, são desconectados.

Nenhum dos indivíduos revelou ter encontrado algum tipo de falhas nos sistemas, salientado que o mesmo funcionou perfeitamente durante o período de tempo em que estiveram com ele. O que se conclui que o protótipo desenvolvido neste projeto se encontra funcional para incorporar um ambiente de vida assistido e para ajudar qualquer pessoa na gestão da sua própria medicação.

Não obstante devemos referir que consideramos que ambos sistemas *Help Medicine* desenvolvidos podem ser melhorados, quer a nível de *design* como a nível funcional. Estas sugestões de melhoria como um trabalho futuro estão descritas no próximo capítulo.

6. Conclusões

Desde os tempos mais remotos que a humanidade tem aproveitado os recursos existentes para melhorar o seu bem-estar e qualidade de vida. É disso exemplo a grande diversidade de dispositivos, mais ou menos evoluídos tecnologicamente, que povoam as nossas vidas e nos auxiliam no quotidiano.

É graças a estes avanços nas tecnologias médicas e farmacêuticas que hoje em dia se pode afirmar que os medicamentos podem curar ou controlar múltiplas doenças ajudando as pessoas a viver ativamente e com uma melhor qualidade de vida durante largos anos. O problema surge quando os sujeitos alvos de medicação são pessoas com necessidades especiais, idosos que vivem sozinhos e não têm qualquer tipo de ajuda para a toma atempada dos respetivos medicamentos, ou então, simplesmente pessoas esquecidas. Muitas das vezes, devido à grande quantidade de medicamentos, os doentes acabam por fazer uma má gestão dos mesmos, sendo que a gestão da medicação se revela uma das maiores preocupações para um doente. Tanto o abuso como a falta de um medicamento podem resultar em situações críticas, potencialmente fatais.

Foi neste contexto que surgiu a necessidade de desenvolver algo que permitisse a estas pessoas uma vida mais autónoma. Neste projeto desenvolveram-se dois protótipos (Aplicação Android e sistema embebido) que alertam para a toma de medicamentos numa data e hora previamente programadas.

A implementação do sistema resultante (*Help Medicine*) foi alcançada com sucesso indo de encontro com os objetivos propostos inicialmente. Foi desenvolvido um protótipo com capacidade para de alertar a toma da medicação aos pacientes. Foi criada uma plataforma física, através da plataforma Arduíno com um tabuleiro de tomas em formato retangular, compartimentado por dias da semana, com aviso sonoro e luminoso.

A aplicação móvel *Help Medicine* permite entre outras funcionalidades, o registo da medicação, o registo de alertas, bem como uma visão geral dos alertas que existem, encomendas de medicamentos, notificações através de *email*, indo de encontro aos objetivos inicialmente propostos.

Os testes de validação realizados com o objetivo de testar a funcionalidade dos protótipos, embora efetuados com um número reduzido de amostra, mostraram que o

dispositivo *Help Medicine*, apesar de ser uma validação de conceito mostra-se funcional e apto para integrar num ambiente de vida de qualquer pessoa.

6.1. Trabalhos Futuros

O projeto desenvolvido nesta dissertação consiste na análise, desenvolvimento e implementação de um protótipo, composto por software e hardware, para alerta de toma de medicamentos. Designamos os mesmos por protótipo uma vez que estes não foram desenvolvidos com intuito comercial.

A título pessoal, considero que o trabalho desenvolvido carece de algumas possíveis melhorias num trabalho futuro, até para que deixe apenas de ser visto como um protótipo e sim como um dispositivo funcional e comercial. As sugestões de melhoria, trabalho futuro, baseiam-se no modo de funcionamento de alerta, no formato e no design do dispositivo.

Como sugestões de trabalho futuro, possamos a enumerar:

- Para que o dispositivo pudesse ser considerado mais fácil de transportar, seria necessária uma redução do seu tamanho. Permitindo assim, que o indivíduo pudesse transportar o dispositivo sempre consigo, evitando assim possíveis esquecimentos de tomas. Para tal, seria necessário idealizar um novo design e utilizar distintos componentes.
- A aplicação móvel exige que o doente, disponha de ligação à Internet. Seria importante que a aplicação fosse também desenvolvida uma em modo *offline* que permitisse criar paciente e criar/editar alerta.
- Ao longo deste projeto foi dada uma maior importância ao facto de se desenvolver um dispositivo funcional menosprezando o *design* do mesmo. No entanto, considero que o design dos sistemas pode e deve ser melhorado num futuro, no que respeita à usabilidade.
- O sistema embebido carece de algumas funcionalidades, como apagar e editar alarmes sem que para tal seja necessário desligar o dispositivo. Com a adição destas funcionalidades o utilizador não teria que

desligar o dispositivo sempre que precisar de alterar ou eliminar um alarme.

- No sistema embebido, também seria interessante haver mais compartimentos no dispositivo. Neste caso, seria ideal que existissem 3 compartimentos, um para de manhã, outro para o almoço e por fim para a medicação da noite. Desta maneira, consideramos, que esta subdivisão facilitava a maneira como o utilizador faz a gestão da toma da sua medicação possibilitando criar 3 vezes mais alertas.
- Ligar os dois sistemas (aplicação móvel e sistema embebido). Seria interessante os dois sistemas trabalharem em conjunto. Neste caso definiríamos os alarmes na aplicação móvel e o sistema embebido alertava o utilizador.

Bibliografia

- [1] A. Campos, «Idosos tomam mais de sete medicamentos por dia, em média», *Público*, 2009.
- [2] Instituto Nacional de Estatística, «Esperança de vida atingiu 80,78 anos à nascença e 19,45 anos aos 65 anos», vol. 2007, pp. 1–8, 2015.
- [3] H. Goulet *et al.*, «Évaluation de la qualité des prescriptions d’antibiotiques dans le service d’accueil des urgences d’un CHU en région parisienne», *Med. Mal. Infect.*, vol. 39, n. 1, pp. 48–54, Jan. 2009.
- [4] M. Hilbert e P. López, «The world’s technological capacity to store, communicate, and compute information», *Science (80-.)*, vol. 332, n. 6025, pp. 60–65, Abr. 2011.
- [5] U. Varshney, «A smart approach to medication management», *Computer (Long. Beach. Calif.)*, vol. 46, n. 1, pp. 71–76, Jan. 2013.
- [6] R. Espanha, «Plano Nacional de Saúde 2011-2016 –Tecnologias de Informação e Comunicação», 2010.
- [7] M. He, «The Evolution and Future Trends of Technology and Information Sciences», em *2009 WASE International Conference on Information Engineering*, 2009, pp. 11–15.
- [8] M. Plimsoll, «Digital Experience Blog | The Future of Banking Technology Is Here», 2018.
- [9] K. R. Lee, «Impacts of Information Technology on Society in the new Century», *Structure*, pp. 1–6, 2002.
- [10] Institute for the Future (IFTF) e Dell Technologies, «The next era of human machine partnerships. Emerging technologies impact on society & work in 2030», p. 24, 2017.
- [11] A. Alturkistani, A. Majeed, J. Car, D. Brindley, G. Wells, e E. Meinert, «Health information technology uses for primary prevention in preventive medicine: a

- scoping review protocol», *BMJ Open*, vol. 8, n. 9, p. e023428, Set. 2018.
- [12] D. C. Carvalho Barra, E. R. Pereira do Nascimento, J. D. J. Martins, G. L. Albuquerque, e A. L. Erdmann, «Evolução histórica e impacto da tecnologia na área da saúde e da enfermagem», *Rev. Eletrônica Enferm.*, vol. 8, n. 3, pp. 422–430, 2009.
- [13] Statista, «Ways to access health apps by patients worldwide 2017 | Statistic». [Em linha]. Disponível em: <https://www.statista.com/statistics/809407/patients-ways-of-accessing-health-apps-worldwide/>. [Acedido: 02-Out-2018].
- [14] Statista, «Greatest market potential of mobile health app categories according to mHealth professionals as of 2016», *Statista*, 2017. [Em linha]. Disponível em: <https://www.statista.com/statistics/655148/market-potential-of-mobile-health-app-categories/>. [Acedido: 02-Out-2018].
- [15] K. Terry, «Prescribing mobile apps: What to consider», *Med. Econ.*, pp. 2–4, 2015.
- [16] K. F. Foreman *et al.*, «Impact of a Text Messaging Pilot Program on Patient Medication Adherence», *Clin. Ther.*, vol. 34, n. 5, pp. 1084–1091, Mai. 2012.
- [17] K. Loudon, «Telemedicine Connects Earthquake-Ravaged Haiti to the World», *Medscape medical news*, 2010. [Em linha]. Disponível em: <https://www.medscape.com/viewarticle/717232>. [Acedido: 02-Out-2018].
- [18] WebMD, «WebMD Mobile Apps». [Em linha]. Disponível em: <https://www.webmd.com/mobile>. [Acedido: 03-Jun-2018].
- [19] Libelium, «MySignals - eHealth and Medical IoT Development Platform». [Em linha]. Disponível em: <http://www.my-signals.com/#buy-mysignals>. [Acedido: 03-Out-2018].
- [20] D. Libelium Comunicaciones, «MySignals SW eHealth and Medical IoT Development Platform Technical Guide», vol. 4.5, p. 305, 2017.
- [21] L. Osterberg e T. Blaschke, «Adherence to Medication», *N. Engl. J. Med.*, vol. 353, n. 5, pp. 487–497, Ago. 2005.
- [22] R. Programming e M. Alert, «MedMinder Pill Dispenser 's Features at a glance».

-
- [23] P. S. Powe, «Pill Dispenser», 1999. [Em linha]. Disponível em: <https://www.medminder.com/pill-dispensers-2/>. [Acedido: 03-Out-2018].
- [24] S. Barker e R. Cole, *Gestão de projeto*. 2014.
- [25] Shams-UI-Arif, Q. Khan, e S. A. K. Gahyyur, «Requirements Engineering Methodologies», *Int. J. Rev. Comput.*, pp. 41–56, 2009.
- [26] H. L. Pinto, «Atividades básicas ao processo de desenvolvimento de Software», *Revista Engenharia de Software*, 2014.
- [27] S. Guerreiro, *Introdução à Engenharia de Software*. 2015.
- [28] R. S. Pressman e B. R. Maxim, *Engenharia de Software. Uma Abordagem Profissional - 8ª Edição*. MCGRAW HILL - ARTMED, 2016.
- [29] R. R. Young, «Recommended requirements gathering practices», *Crosstalk*, vol. 15, n. 4, pp. 9–12, 2002.
- [30] T. Giovanella, «Levantamento de Requisitos», *Ciências da Computação*, pp. 1–33, 2013.
- [31] C. E. Vazquez e G. Siqueira, *Engenharia de Requisitos - software orientado ao negócio*. 2016.
- [32] AEP, «Guia Prototipagem», 2017.
- [33] I. Sommerville, *Engenharia de Software*, 9.^a ed. 2011.
- [34] M. N. | H. O'Neill, *Fundamental de UML - Informática - Sistemas de Informação & Engenharia de Software - FCA*, 3ª Edição. 2004.
- [35] ExperienceUX, «What is wireframing | Experience UX», *experience UX.co.uk*, 2015. [Em linha]. Disponível em: <http://www.experienceux.co.uk/faqs/what-is-wireframing/>. [Acedido: 08-Jul-2017].
- [36] B. Hallgrímsson, *Prototyping and modelmaking for product design*. 2012.
- [37] Arduino, «Pushbutton», 2015. [Em linha]. Disponível em: <https://www.arduino.cc/en/tutorial/pushbutton>. [Acedido: 10-Out-2018].
- [38] W. E. Ltd, «Friedland D642 Spare Push Button». [Em linha]. Disponível em:
-

- <https://www.wesco.ie/products/p-d642.html>. [Acedido: 10-Out-2018].
- [39] H. Sakai e T. Kawamura, «Light-emitting diode», Ago. 1986.
- [40] H. Mattede, «O que é um led?» [Em linha]. Disponível em: <http://creeled.weebly.com/o-que-e-um-led.html>. [Acedido: 09-Out-2018].
- [41] Arduino, «LiquidCrystal». [Em linha]. Disponível em: <https://www.arduino.cc/en/Tutorial/HelloWorld>. [Acedido: 10-Out-2018].
- [42] Amazon, «I2C Serial LCD Module Display». [Em linha]. Disponível em: <https://www.amazon.com/SainSmart-Serial-Module-Display-Arduino/dp/B00813HBEQ>. [Acedido: 10-Out-2018].
- [43] Huinfinito, «Buzzer». [Em linha]. Disponível em: <http://www.huinfinito.com.br/arduino/969-modulo-sensor-buzzer-5v.html>. [Acedido: 10-Out-2018].

Anexo A – Análise de requisitos

Descrição casos de uso e diagramas de sequência

Tabela 10 – Descrição do caso de uso “Editar utilizador”.

Nome	Editar utilizador
Objetivo	Este caso de uso tem como objetivo principal editar um utilizador da aplicação.
Atores	Utilizador
Pré-condição	Nenhuma
Pós-condição	Alterar dados do utilizador no sistema.
Caminho Principal	<ol style="list-style-type: none">1. O ator seleciona a opção “Perfil” + nome do utilizador.2. O sistema exibe o formulário para editar o utilizador.3. Se o ator seleciona a opção “Escolher foto”;4. O sistema atualiza a foto no formulário.5. O ator altera os dados que pretende (nome, idade, médica de família, problemas de saúde).6. O ator clica no botão guardar para confirmar e atualizar os dados.7. O sistema altera os dados do utilizador.

	8. O sistema encaminha o utilizador para o menu e retorna mensagem de sucesso de alteração.
Caminho alternativo	<p>4. O sistema exhibe as opções, “Tirar foto” ou “Escolher foto da Galeria”.</p> <p>a) Se o ator optar por “Tirar foto”</p> <p>4.1.a.1. O sistema abre a câmara do dispositivo;</p> <p>b) Se o ator optar por “Escolher foto da Galeria”</p> <p>4.1.b.1. O sistema abre a galeria do dispositivo;</p> <p>c) Voltar para o ponto 4 do caminho principal.</p>
Caminho de exceção	<p>3. a) O sistema pode não mostrar as opções para a foto, não foram ativadas as permissões.</p> <p>6. a) O ator não preencheu os dados corretamente, retorna para o ponto 5 do caminho principal, sinalizando quais os campos que não foram preenchidos de forma correta.</p>
Suplementos ou adornos (requisitos segurança)	<p>Fazer vários testes um de verifique:</p> <p>a) Se o utilizador é capaz de alterar os dados pretendidos.</p>

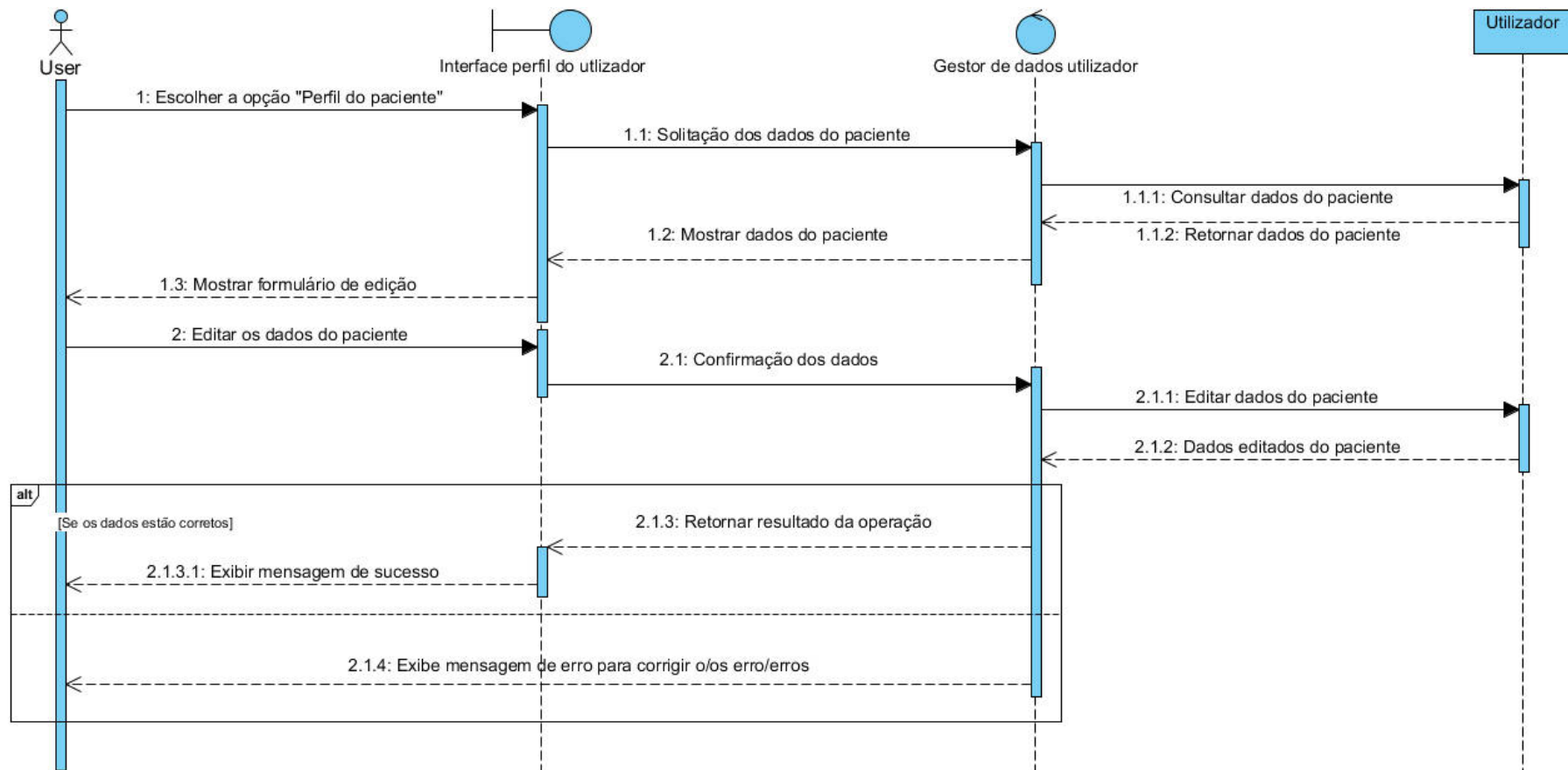


Figura 49 - Sequência de operações – “Editar Utilizador”.

Tabela 11 - Descrição do caso de uso “Editar alerta medicamentos”.

Nome	Editar alerta medicamentos
Objetivo	Este caso de uso tem como objetivo selecionar qual o medicamento que o utilizador pretende ser alertado.
Atores	Utilizador
Pré-condição	Utilizador no seu perfil.
Pós-condição	Alterar dados do alerta no sistema.
Caminho Principal	<ol style="list-style-type: none">1. O ator seleciona a opção “Alertas”.2. O sistema exhibe o formulário do alerta.3. O ator altera os dados pretendidos, tais como, hora, data de início, data de fim, dosagem e ativo.4. O sistema altera os dados do alerta.5. O sistema retorna mensagem de sucesso da alteração.
Suplementos ou adornos (requisitos segurança)	Fazer vários testes um de verifique: <ol style="list-style-type: none">a) Se o utilizador consegue alterar um alerta.

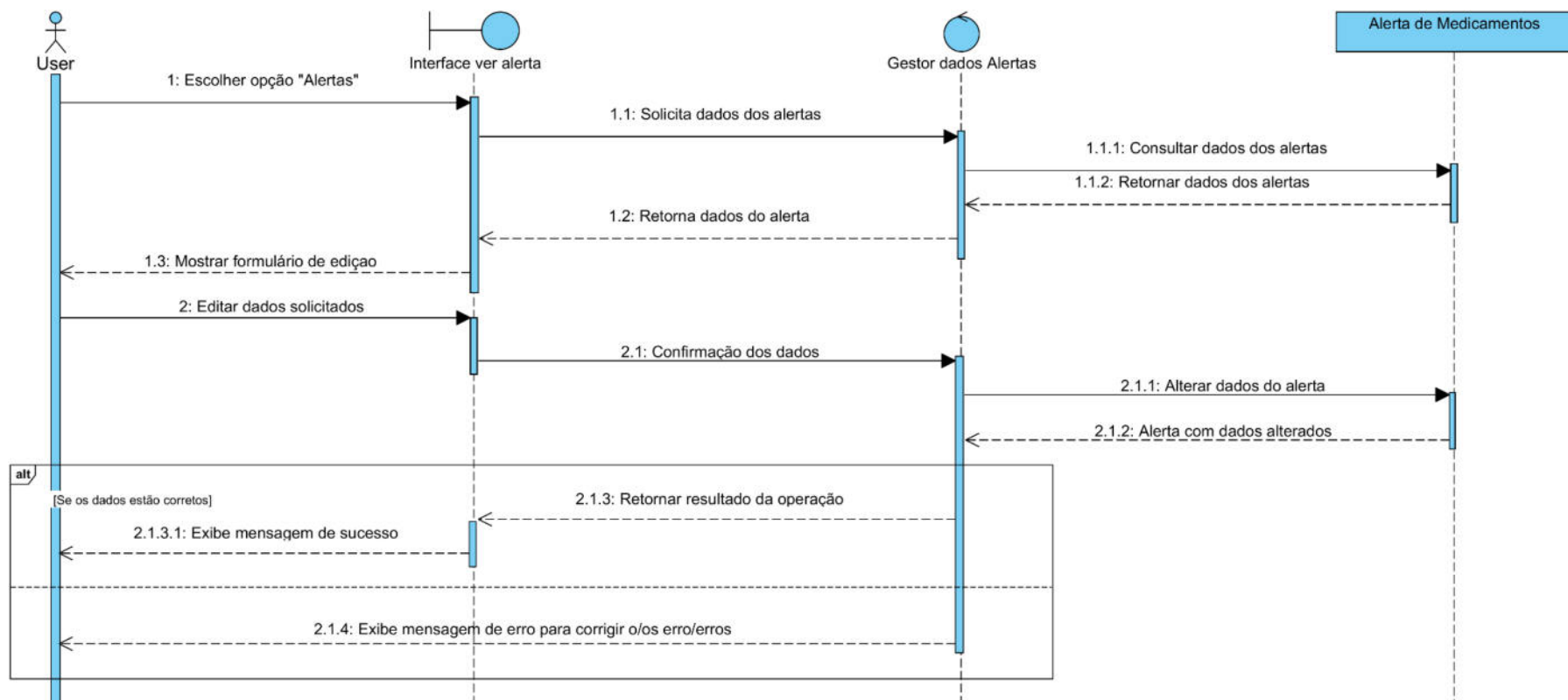


Figura 50 - Sequência de operações – “Editar alerta medicamentos”.

Tabela 12 - Descrição do caso de uso “Eliminar alerta medicamento”

Nome	Eliminar alerta medicamento
Objetivo	Este caso de uso tem como objetivo eliminar qual o medicamento que o utilizador já não pretende ser alertado.
Atores	Utilizador
Pré-condição	Utilizador no seu perfil
Pós-condição	Excluir alerta do utilizador do sistema.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a opção Alertas. 2. O ator seleciona a opção “Apagar alerta”. 3. O ator confirma a opção de apagar alerta. 4. O sistema exclui os dados do alerta. 5. O sistema retorna mensagem de sucesso da eliminação.
Caminhos Alternativos	3. O ator clicou em “Não” na caixa de diálogo de confirmação para excluir alertas, retorna para o passo 1 do caminho principal.
Suplementos ou adornos (requisitos segurança)	<p>Fazer vários testes um de verifique:</p> <ol style="list-style-type: none"> a) Se o utilizador consegue excluir um alerta de medicamento.

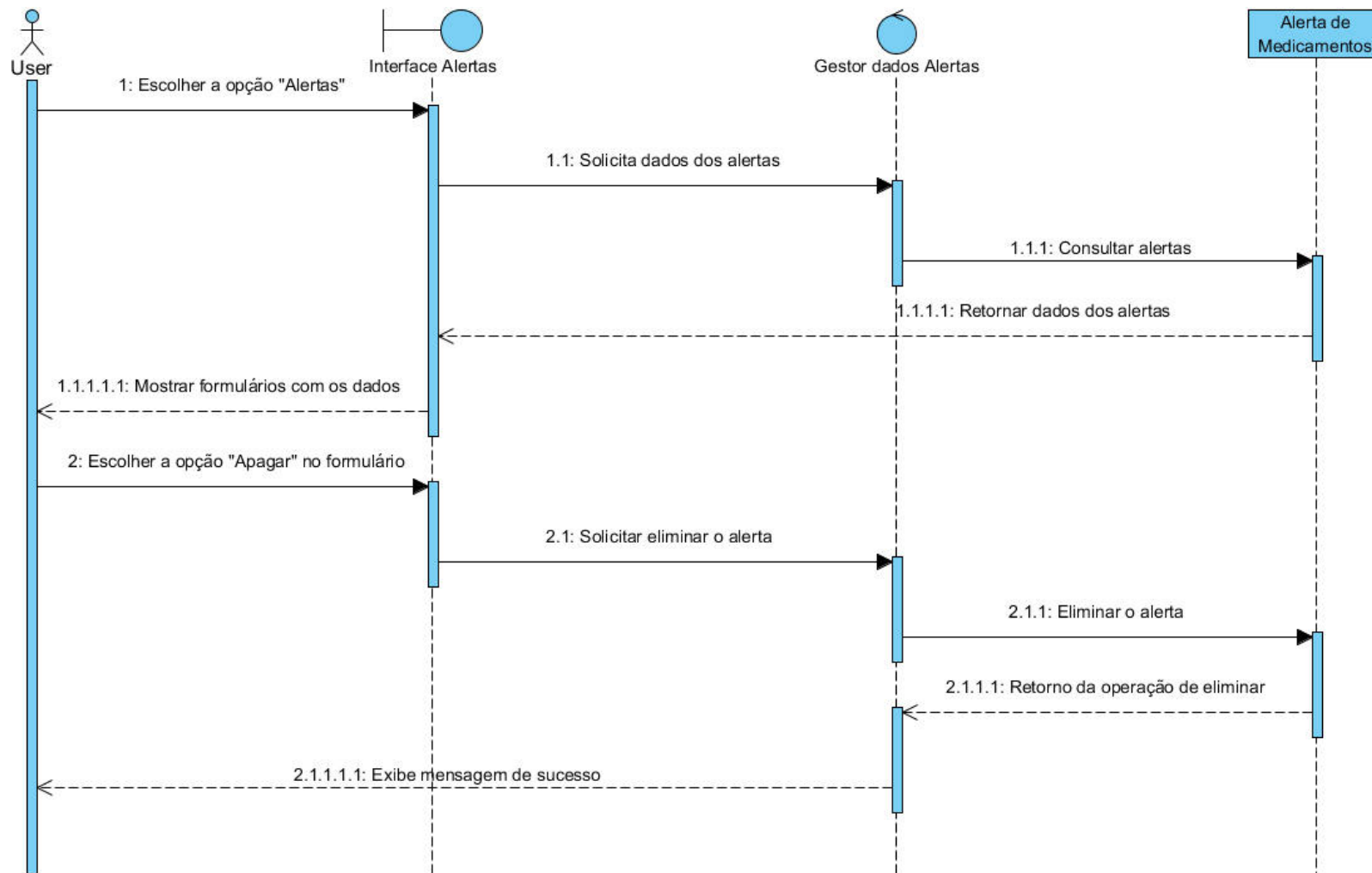


Figura 51 - Sequência de operações – “Eliminar alerta medicamentos”

Tabela 13 - Descrição do caso de uso “Consultar alerta medicamento”

Nome	Consultar alerta medicamento
Objetivo	Este caso de uso tem como objetivo consultar os alertas de medicamentos.
Atores	Utilizador
Pré-condição	Utilizador no seu perfil.
Pós-condição	Visualização dos alertas do utilizador no sistema.
Caminho Principal	<ol style="list-style-type: none">1. O ator seleciona em “Alertas”2. O sistema exibe os alertas do utilizador.3. O ator clicar na opção ara voltar ao menu.4. O sistema encaminha o utilizador para o menu.
Caminhos de exceção	<ol style="list-style-type: none">2. a) Não existem alertas para aquele utilizador, o sistema exibe uma mensagem: “Não há alertas para exibir.”
Suplementos ou adornos (requisitos segurança)	Fazer vários testes um de verifique: <ol style="list-style-type: none">a) Se o utilizador consegue ver os alertas.

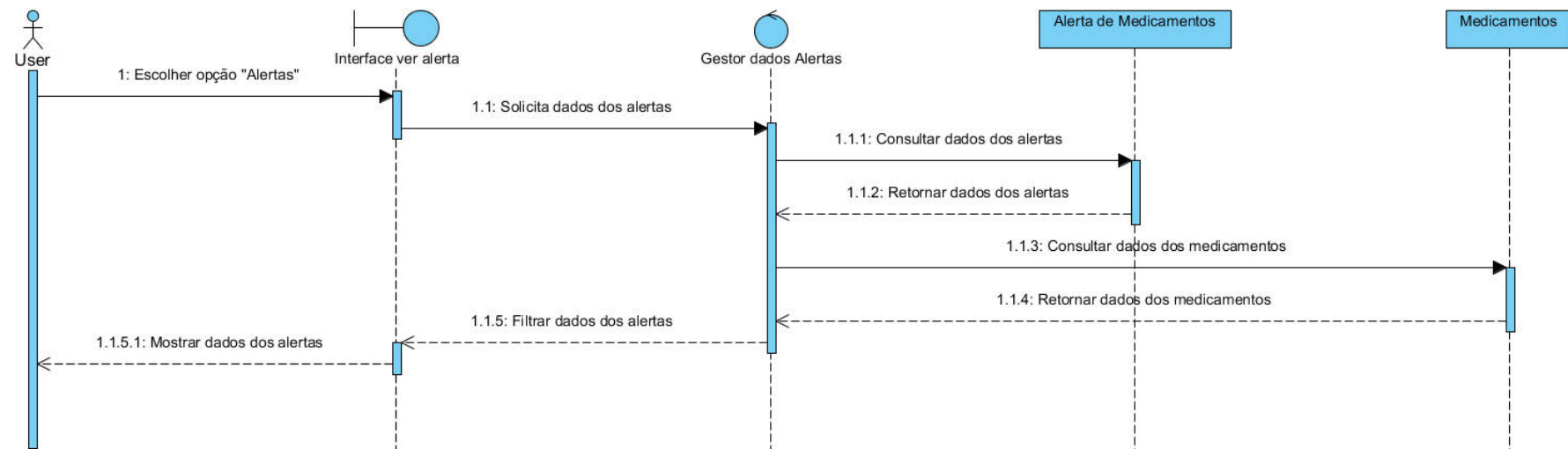


Figura 52 - Sequência de operações – “Consultar alerta medicamento”

Tabela 14 - Descrição do caso de uso “Inserir medicamento”

Nome	Inserir medicamento
Objetivo	Este caso de uso tem como objetivo inserir medicamentos.
Atores	Utilizador
Pré-condição	Nenhuma
Pós-condição	Registrar medicamento no sistema
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a opção “Medicamentos”. 2. O ator seleciona a opção “Novo medicamento” 3. O sistema exibe um formulário para preenchimento de um novo medicamento. 4. O ator preenche o nome do medicamento. 5. O ator clica no botão “Guardar” para confirmar e enviar os dados. 6. O sistema guarda os dados do medicamento 7. O sistema encaminha o utilizador para o submenu “Medicamentos” e retorna mensagem de sucesso de inserção.
Caminhos de exceção	<ol style="list-style-type: none"> 5. a) O ator não preencheu os dados corretamente, retorna para o ponto 4 sinalizando que o campo não foi preenchido corretamente.
Suplementos ou adornos (requisitos segurança)	<p>Fazer vários testes um de verifique:</p> <ol style="list-style-type: none"> b) Se o utilizador consegue inserir um medicamento.

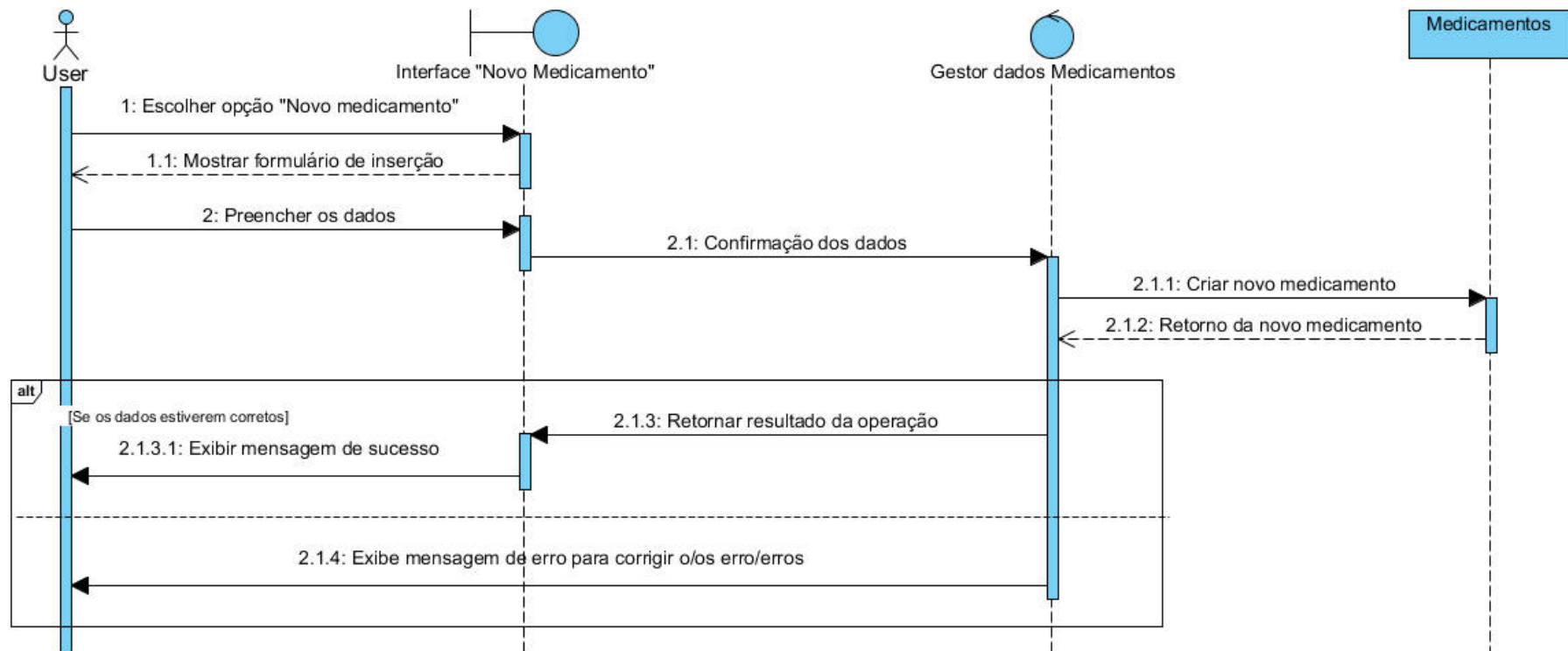


Figura 53 - Sequência de operações – “Inserir medicamento”

Tabela 15 - Descrição do caso de uso “Consultar medicamentos”

Nome	Consultar medicamentos
Objetivo	Este caso de uso tem como objetivo mostrar uma lista de todos os medicamentos existentes.
Atores	Utilizador
Pré-condição	Não tem.
Pós-condição	Visualização dos medicamentos.
Caminho Principal	<ol style="list-style-type: none">1. O ator seleciona a opção “Medicamentos”2. O ator seleciona a opção “Ver medicamentos”.3. O sistema exibe uma lista com os nomes dos medicamentos.
Caminhos Alternativos	<ol style="list-style-type: none">3. O ator pretender pesquisar medicamentos<ol style="list-style-type: none">a) O ator clica na opção “Pesquisar medicamentos”;b) O ator insere o nome do medicamento que deseja pesquisar;c) o sistema mostra o medicamento que pretende.
Caminho de exceção	<ol style="list-style-type: none">3. a) No caminho alternativo, o ator pesquisa o nome do medicamento que não existe, retorna botão para adicionar um novo medicamento.
Suplementos ou adornos (requisitos segurança)	Fazer vários testes onde se verifique: <ol style="list-style-type: none">a) Que a pesquisa de medicamentos funciona sem limitações.

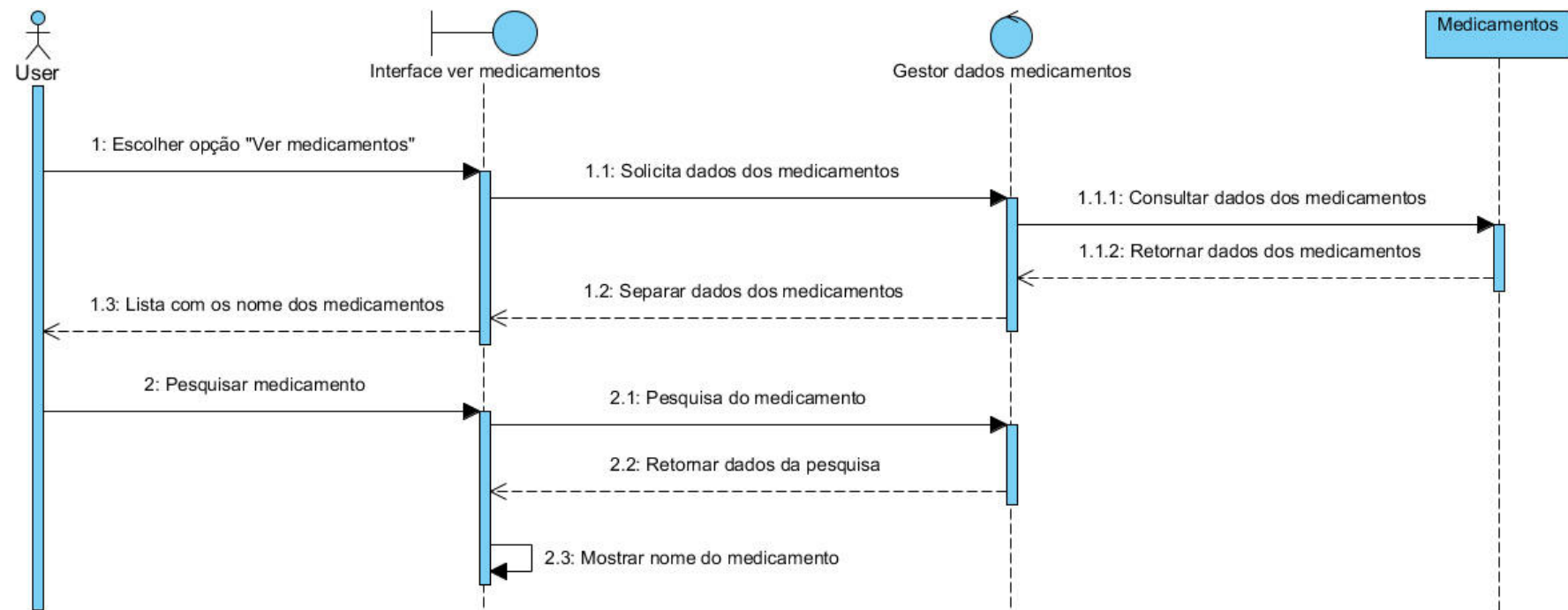


Figura 54 - Sequência de operações – “Consultar medicamentos”

Tabela 16 - Descrição do caso de uso “Inserir Farmácias”

Nome	Inserir Farmácias
Objetivo	Este caso de uso tem como objetivo inserir farmácias.
Atores	Utilizador
Pré-condição	Nenhuma
Pós-condição	Registar farmácia no sistema
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a opção “Farmácias”. 2. O ator seleciona a opção “Adicionar farmácia” 3. O sistema exibe um formulário para preenchimento de uma nova farmácia. 4. O ator preenche o nome e <i>email</i> da farmácia. 5. O ator clica no botão “Guardar” para confirmar e enviar os dados. 6. O sistema guarda os dados da farmácia. 7. O sistema encaminha o utilizador para o submenu “Farmácias” e retorna mensagem de sucesso de inserção.
Caminhos de exceção	<ol style="list-style-type: none"> 6. a) O ator não preencheu os dados corretamente, retorna para o ponto 4 sinalizando quais os campos que não foram preenchidos corretamente.
Suplementos ou adornos (requisitos segurança)	<p>Fazer vários testes um de verifique:</p> <ol style="list-style-type: none"> a) Se o utilizador consegue inserir uma farmácia.

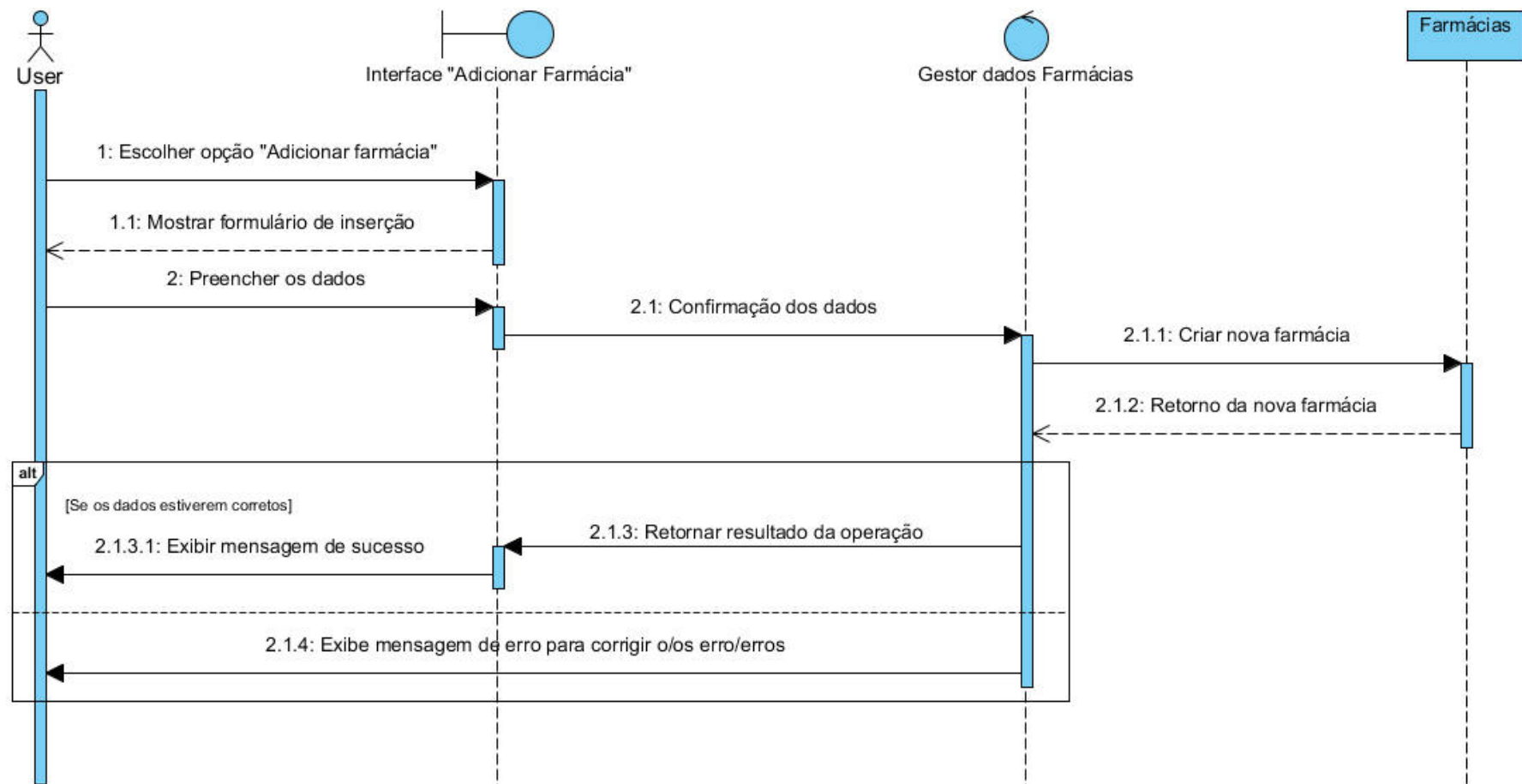


Figura 55 - Sequência de operações – “inserir Farmácia”

Tabela 17 – Descrição do caso de uso “Consultar farmácias”

Nome	Consultar farmácias
Objetivo	Este caso de uso tem como objetivo mostrar uma lista de todos os medicamentos existentes.
Atores	Utilizador
Pré-condição	Não tem.
Pós-condição	Visualização dos medicamentos.
Caminho Principal	<ol style="list-style-type: none"> 4. O ator seleciona a opção “Farmácias” 5. O ator seleciona a opção “Ver farmácias”. 6. O sistema exibe uma lista com os nomes das farmácias.
Caminhos Alternativos	<ol style="list-style-type: none"> 4. O ator pretender pesquisar farmácias <ol style="list-style-type: none"> a) O ator clica na opção “Pesquisar farmácias”; b) O ator insere o nome da farmácia que deseja pesquisar; c) o sistema mostra a farmácias que pretende.
Caminho de exceção	<ol style="list-style-type: none"> 4. a) No caminho alternativo, o ator pesquisa o nome da farmácia que não existe, retorna botão para adicionar uma nova farmácia.
Suplementos ou adornos (requisitos segurança)	<p>Fazer vários testes onde se verifique:</p> <ol style="list-style-type: none"> b) Que a pesquisa de farmácias funciona sem limitações.

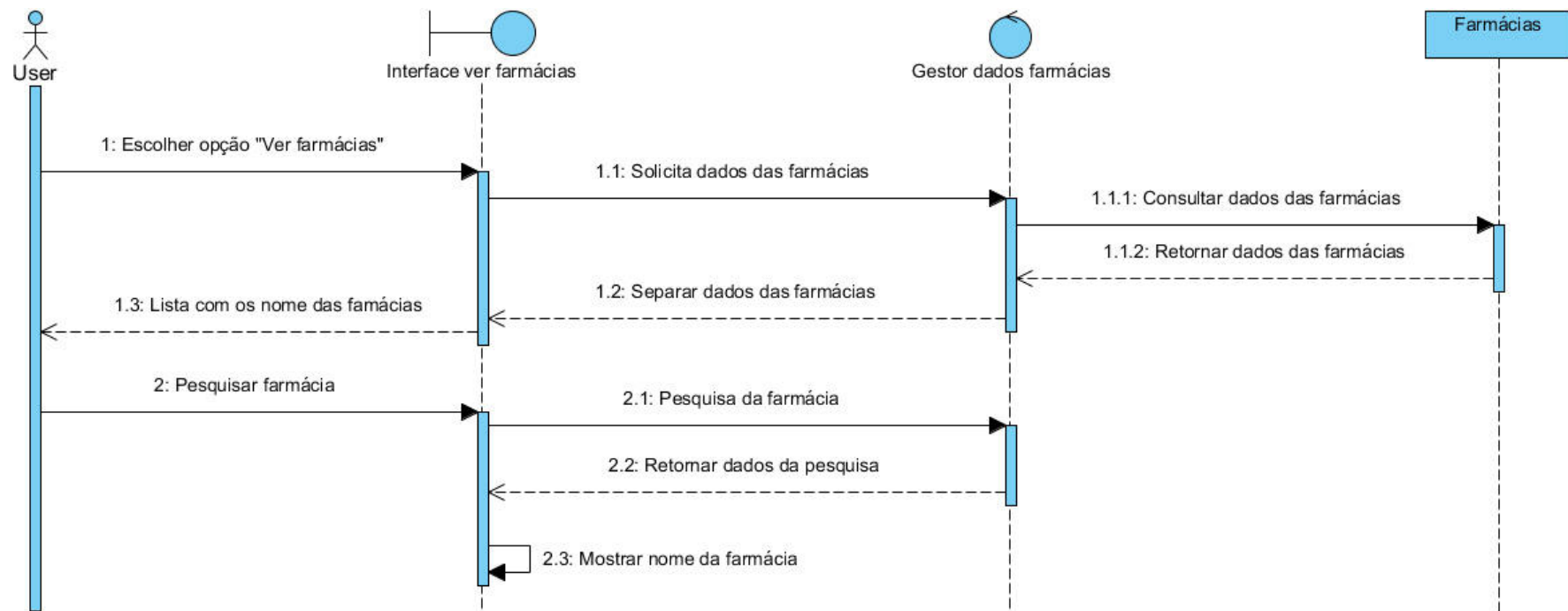


Figura 56 - Sequência de operações – “Consultar farmácias”

Nome	Consultar encomendas
Objetivo	Este caso de uso tem como objetivo mostrar uma lista das encomendas existentes.
Atores	Utilizador
Pré-condição	Utilizador no perfil.
Pós-condição	Visualização das encomendas.
Caminho Principal	<ol style="list-style-type: none"> 1. O ator seleciona a opção “Farmácias” 2. O ator seleciona a opção “Detalhes das encomendas”. 3. O sistema exibe uma lista com as encomendas.
Caminhos Alternativos	<ol style="list-style-type: none"> 5. O ator pretender cancelar a encomenda <ol style="list-style-type: none"> a) O ator clica na opção “cancelar encomenda”; b) O sistema exibe mensagem de sucesso. c) Voltar ao ponto 3 do caminho principal.
Suplementos ou adornos (requisitos segurança)	<p>Fazer vários testes onde se verifique:</p> <ol style="list-style-type: none"> c) Se o cancelamento de encomenda ocorre.

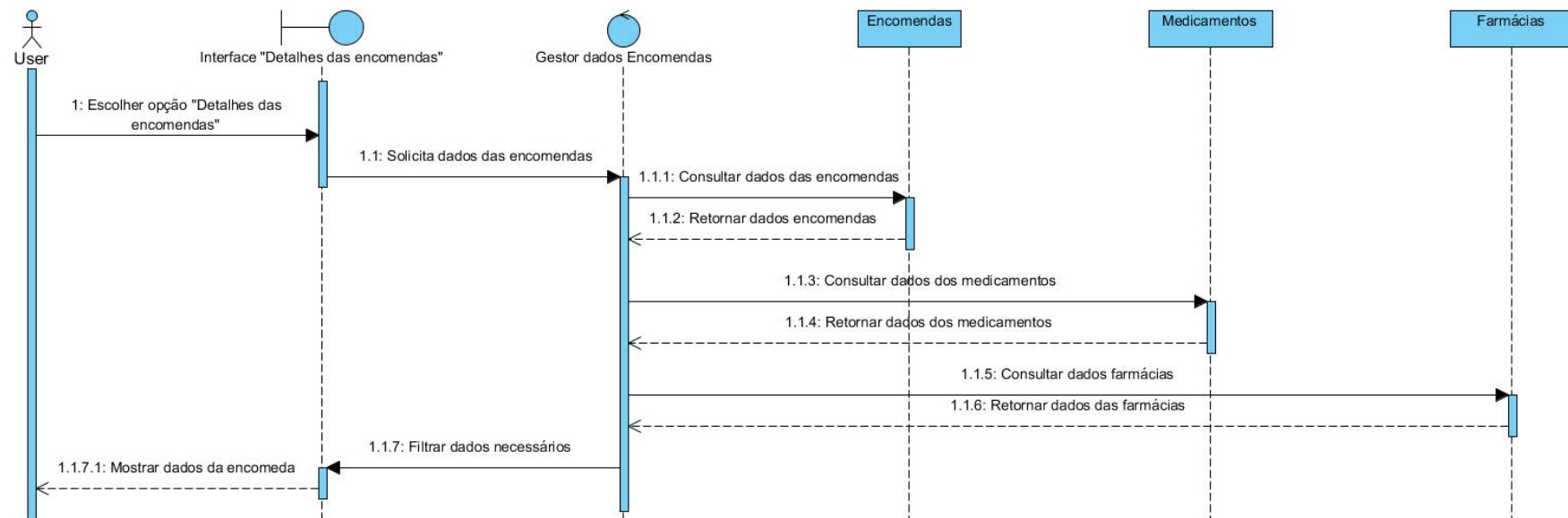


Figura 57 - Sequência de operações – “Consultar encomendas”

Anexo B – Wireframes



Figura 58 – Wireframe utilizadores da aplicação Help Medicine

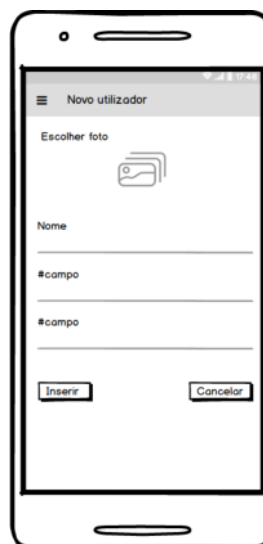


Figura 59 - Wireframe inserir utilizadores da aplicação Help Medicine



Figura 60 - Wireframe menu dos utilizadores da aplicação Help Medicine

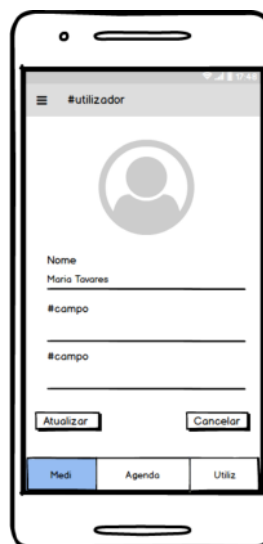


Figura 61 - Wireframe perfil do utilizador da aplicação Help Medicine



Figura 62 - Wireframe inserir alerta da aplicação Help Medicine



Figura 63 - Wireframe inserir medicamento da aplicação Help Medicine

Anexo C – Formulário de validação do *Help Medicine*

Género: Feminino ☐ Masculino ☐

Data de Nascimento: ____/____/____

Toma medicação diária?

Sim ☐ Não ☐

Se sim...

Quantos medicamentos? ____

A posologia da sua medicação inclui que partes do dia?

Manhã ☐ Meio-dia ☐ Tarde ☐ Noite ☐

Durante o tempo de utilização do *Help Medicine*, alguma vez se esqueceu da toma da sua medicação?

Sim ☐ Não ☐

Sente que o *Help Medicine*, de alguma forma o ajudou?

Sim ☐ Não ☐

Considera que o *Help Medicine* o pode ajudar numa vida mais autónoma?

Sim ☐ Não ☐